MEMORANDUM
RM-5544-PR
MAY 1968

THE LOGISTICS COMPOSITE MODEL: AN OVERALL VIEW

Captain R. R. Fisher, W. W. Drake, J. J. Delfausse, A. J. Clark, and A. L. Buchanan

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA · CALIFORNIA

112

MEMORANDUM
RM-5544-PR
MAY 1968

THE LOGISTICS COMPOSITE MODEL: AN OVERALL VIEW

Captain R. R. Fisher, W. W. Drake, J. J. Delfausse,
A. J. Clark and A. L. Buchanan

This research is supported by the United States Air Force under Project RAND — Contract No. F44620-67-C-0045 — monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. RAND Memoranda are subject to critical review procedures at the research department and corporate levels. Views and conclusions expressed herein are nevertheless the primary responsibility of the author, and should not be interpreted as representing the official opinion or policy of the United States Air Force or of The RAND Corporation.

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90406

PREFACE

This Memorandum describes a computer model for simulating a composite of operations and support functions at an Air Force base. Identified as the Logistics Composite Model (L-COM), its development has been a joint effort of personnel at Headquarters, Air Force Logistics Command, and The RAND Corporation.

Presented here is an overall description of the model and its use. Separate reference manuals will describe technical details concerning input data preparation, program logic, and interpretation of outputs.

Captain R. R. Fisher, W. W. Drake, and J. J. Delfausse were with the Air Force Logistics Command when this research was performed. A. J. Clark and A. L. Buchanan are with The RAND Corporation. Mr. Clark is a consultant.

SUMMARY

This Memorandum describes a model for simulating overall operations and support functions at an Air Force base. Identified as the Logistics Composite Model (L-COM), it is applicable to a variety of planning studies concerned with base level functions. The description is oriented toward how the operational environment is represented, what decision processes are included, and how the model may be used in various kinds of planning studies.

L-COM consists of three main programs: a preprocessor, a simulation program, and a postprocessor. The preprocessor translates data provided on specialized forms designed for user convenience into data forms the simulation program requires. It also generates sortie requirements in accordance with a specified flying program.

The simulation program, driven by data from the preprocessor, represents flight and base support processes in response to mission requirements. The logic of the model replicates the flying of aircraft; the accomplishment of servicing tasks such as refueling and weapons loading; the incurrence of malfunctions; the accomplishment of flight-line aircraft maintenance; the repair of components in base repair shops; the utilization and interaction of resources in the demand process; the changes in resource availability according to shift policies; and other facets of the overall base operation.

As the simulation program operates, it presents results in a periodic performance summary report. At the user's option, the program may produce several kinds of status reports. Using results of the simulation, the postprocessor presents summary statistics in graphical form and, for selected aircraft, displays of the processes incurred by the airplane and components removed for shop repair.

In normal operation, the preprocessor is first used to develop a data base for the simulation. The simulation program may then be run many times, with systematic changes being made in the data base according to particular study objectives. The postprocessor may be used after each simulation to provide results in a form convenient for analysis.

The model has two unique features. One is the task network that describes base processes to be simulated by identifying particular tasks and the sequence for accomplishing them. Input data prescribe durations and resource requirements for each task. By providing network and related data, the user exercises direct control over the environmental representation included in the simulation.

The other unique feature consists of embedded decision routines that help determine a best mix of resources to support a prescribed flying program. During a simulation, these routines determine whether given performance goals are being met and, if not, the routines use a cost-effectiveness criterion to augment resource levels selectively until desired performance objectives are attained.

The model requires a computer with an internal memory of at least 65,000 words of 36-bit length or equivalent. A typical problem requires from 1-1/2 to 2 minutes of computer time to simulate a day's worth of base operations involving 1500 different tasks. Since the model is written in SIMSCRIPT, almost any computer of sufficient size may be used.

To validate L-COM, it is planned to use data and experience gained from AFLC Project PACER SORT. Results from both the model and the field test will be compared for a number of key performance measures. If a sufficiently close match is obtained, the model may be used with increased confidence for other applications.

Several extensions and refinements are planned for L-COM. A repair-level decision model is being developed to enable initial determinations of repair policies and associated resource requirements. The simulation program may be extended to include representations of conflicting and defined maintenance, shop cannibalization, weather effects, and multibase operat ns. The embedded decision processes may also be refined in several ways. Finally, data bank and on-line simulation concepts may be employed to facilitate the use of the model in some planning studies.

Present and future versions of L-COM should have significance for logistics requirements studies in support of contingency deployments and determinations of preferred repair policies, as well as associated resource requirements studies for weapon systems being designed. In addition, the model may be used in any problem involving appreciable interaction among the many functions accomplished at an Air Force base.

ACKNOWLEDGMENTS

The existence of the Logistics Composite Model described here is due largely to the interest and support of Major General W. F. Veal who, as Director of Plans and Programs, AFLC, and Director of Project PACER SORT, assigned AFLC personnel to develop the model. Murray A. Geisler, head of the Logistics Department, also provided valuable support by authorizing the use of RAND Corporation personnel and facilities.

The authors gratefully acknowledge the technical contributions of Captain Neil Carscadden of the Royal Canadian Air Force on exchange duty with the USAF at Headquarters AFLC, who provided significant assistance in later model formulation and programming.

Preceding Page Blank

CONTENTS

I. INTRODUCTION

From management's point of view, the operation of an Air Force base encompasses most of the functions of a large business as well as those of a military organization. Included, in particular, are planning functions concerned with the acquisition and efficient use of the many kinds of resources involved in base operations. Due to the complex interrelationships that exist in using resource mixes to accomplish a variety of mission requirements, planning studies of significant scope have been very difficult to perform. To help alleviate this difficulty, several kinds of computer-based analysis techniques have been developed in recent years. One considerably important technique is simulation--the representation of base functions in terms of a model of the environment.

In general, computerized simulation models enable a planner to visualize how a base operates under actual or postulated conditions without disturbing the operation of the base itself. Such a model serves as a kind of laboratory in which an analyst can experiment with and try out new concepts governing base operations and resource utilization. It enables him to determine how base processes are interrelated and affected by postulated changes in the way they are accomplished. By this means, preferred solutions to base management problems may be determined, together with estimates of accrued benefits upon implementation.

This Memorandum describes a model for simulating overall base operations and logistical support functions. Identified as the Logistics Composite Model (L-COM), it is applicable to many planning studies concerned with base-level functions. The description is oriented toward how the operational environment is represented, what decision processes are included, and how the model may be used in various kinds of planning studies.

BACKGROUND

In the past several years, a number of simulation models addressed to weapon system operation and support at Air Force bases have been constructed.[*] Each model has design characteristics that facilitate and usually constrain its use to particular classes of study objectives. In general, these distinguishing features are the way of representing the base environment and the level of detail in simulating various base functions such as operations, maintenance, and supply. Technically, the models are further distinguished by the kinds of input data required, the language in which the computer programs are written, the kinds of computers upon which the models can operate, and the types of output reports produced.

The Logistics Composite Model was also developed to address a particular class of study objectives--that associated with Project PACER SORT (formerly, Project LOGGY SORT). This project, sponsored by the Air Force Logistics Command, concerned the test and evaluation of revised repair and maintenance policies to support tactical aircraft in both peacetime and contingency operations. The main project objective was to determine, from a cost-effectiveness point of view, the best mix of base and depot level repair, with associated requirements for maintenance personnel, ground support equipment, repair parts, transportation, communications and other supporting resources.

To accomplish part of the project objectives, a field test was conducted in South Vietnam during the first six months of 1967. The test involved the support, under combat conditions, of two squadrons of F-4C aircraft with a greater degree of depot level repair of components than prescribed by previous maintenance policies. Included in the test was a control group of two other squadrons operating with similar mission requirements but supported by extensive amounts of base repair according to traditional policies. A comprehensive data collection program was used to measure and evaluate the effects of the two maintenance concepts upon operations and overall logistics

[*]Three such models have been developed at The RAND Corporation. They are described in the following: A. S. Ginsberg and B. A. King, *Base Operations-Maintenance Simulator*, The RAND Corporation, RM-4072, August 1964; T. C. Smith, *SAMSOM: Support-Availability Multi-System Operations Model*, The RAND Corporation, RM-4077-PR, June 1964; and B. J. Voosen, *PLANET: Planned Logistics Analysis and Evaluation Technique*, The RAND Corporation, RM-4950-PR, January 1967.

(671112)

support. Results of the test experience are contained in a final report published on 30 June 1967.*

From the start of Project PACER SORT, a number of limitations of the field test in satisfying the basic study objectives were recognized. First, the test involved F-4C aircraft whose supporting resources (such as AGE--aerospace ground equipment--and repair parts) had already been designed and procured under previous maintenance concepts of maximum base self-sufficiency; this severely constrained possible changes in repair policies and largely prevented any savings in cost and/or increases in operational effectiveness. Second, data collected from the field test represented the operation of a particular type of aircraft in a particular operational and management environment. The usefulness of this data and its results for other types of aircraft operating in other environments and in future time frames was doubtful. Third, due to the size, complexity, and limited duration of the field test, it was realistically impossible to test and evaluate gradations in repair level policies, thereby precluding the determination of the best mix of base and depot level repair.

To overcome these limitations and others, simulation was accepted as a means for augmenting and extending the field test results. We hoped that a simulation model could not only replicate the field test environment in order to extend test results over longer operational time periods, but also apply to other aircraft, operational and logistical support environments, postulated maintenance and supply policies, and conditions prevailing in current and future time frames not encompassed by the field test itself. In general, a simulation model would accomplish many of the study objectives without incurring the considerable expenses for the variety of field tests that would otherwise be indicated.

So a team of analysts and programmers was established to develop a simulation model for the PACER SORT project. The team, consisting of personnel from AFLC and The RAND Corporation, began the development

**Project PACER SORT Special Overseas Repair Test, Final Report,* Headquarters Air Force Logistics Command (67 MCGF-106).

effort at RAND in November 1966. In March 1967, the program was transferred to Headquarters AFLC in Dayton, Ohio, to take advantage of the larger capacity of its UNIVAC 1107 computer. The Logistics Composite Model described here is the result of the joint AFLC-RAND development program.

SCOPE

Essentially, the current L-COM model (MOD 1) can simulate a composite of aircraft operations and main supporting functions at one Air Force base. In particular, it contains representations of flying operations for a mix of aircraft and sortie types, flight-line and shop repair processes for both aircraft and components, and supply functions in support of maintenance. Classes of resources explicitly identified (by line item within each class) in the model include aircraft, personnel, spare parts, AGE, and facilities. In general, L-COM permits representations of all basic functions involved in aircraft operations and support in varying levels of detail as prescribed by the user.

With minor programming changes, particularly in the output report formats, the model can simulate a missile weapon system deployed in a region but subject to common support. With other changes, it can simulate the operation of a depot repair and overhaul facility. To apply the model in these and other areas, the user must develop corresponding input data and make appropriate alterations in the output reports.

Aside from simulating base operations, the model contains decision logic for determining which resource items are degrading system performance the most, and determining how much to increase these item levels to maintain a prescribed effectiveness. The use of this logic for a given model run is at the user's discretion. Except for this logic, there is no representation of supply replenishment policies other than a one-for-one replacement of items lost to the base through condemnation or return to depot.

OVERALL STRUCTURE

The Logistics Composite Model (MOD 1) consists of three main computer programs: a preprocessor, a simulation model, and a postprocessor. Figure 1 shows how these programs interrelate in the overall operation of the model.

The preprocessing or input program has two functions:

1. It serves as a translator to reduce or reformat data provided on specialized forms designed for user convenience into data forms required by the simulation model.
2. It generates sortie requirements for a flying program specified by the user, who defines the requirements in terms of missions, with varying numbers of sorties per mission, to be accomplished at specified times in the simulation.

In translating user-provided data into formats required by the simulation model, the preprocessor edits the data for errors and completeness. When the preprocessor detects errors or inconsistencies, it prints appropriate error messages. In some cases where ambiguities are found, assumptions are made about what the user intended, and a message is provided about what was found and what assumption was made. In more serious cases, where such an assumption cannot be made, further processing is discontinued until the omission is corrected.

The preprocessor outputs two classes of data. The first class (initialization data) describes the environment to be simulated and prescribes initial values for resource levels, reliability factors, policy parameters, and other elements. The second class (exogenous events) describes mission requirements in terms of takeoff times, number of airplanes needed, types of sorties, and other data pertaining to the missions.

Driven and controlled by data from the input program, the simulation model provides a representation of flight and base support processes in response to mission requirements. The logic of the model replicates the flying of aircraft; the accomplishment of servicing tasks such as refueling, weapons loading, etc.; the incurrence of malfunctions; the accomplishment of flight-line aircraft maintenance;

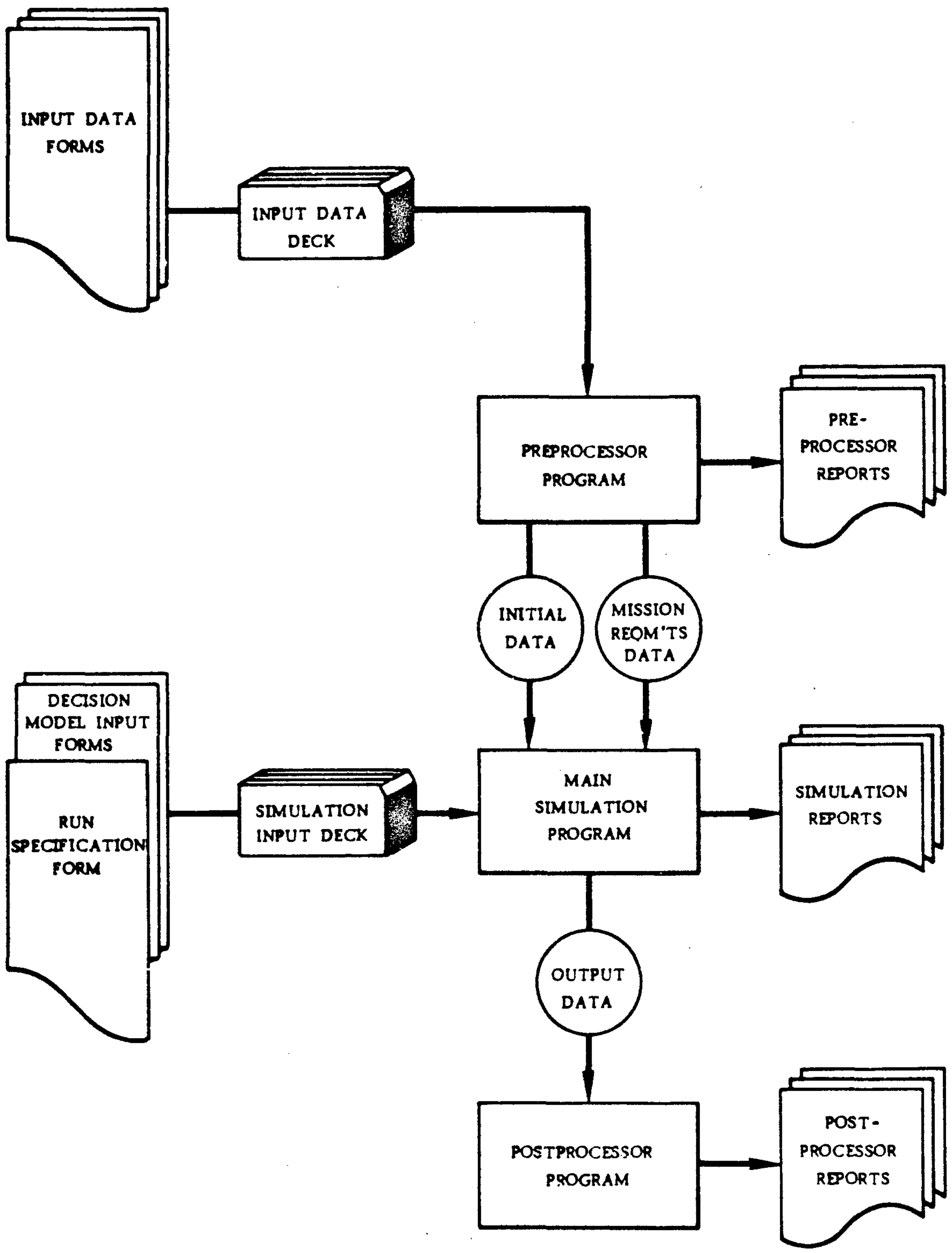


Fig. 1 -- Overall structure of the Logistics Composite Model (MOD 1)

the repair of components in base repair shops; the utilization and interaction of resources in the demand process; changes in resource availability according to shift policies; and other facets of the overall base operation.

In general, user-provided input data determine the degree of detail for base processes. The processes and the way they interrelate are described in task networks, which identify the processes that can be done in parallel (simultaneously) and those that must be accomplished sequentially. For each network process or task, the user provides data determining the task duration and required kinds and amounts of resources. By so doing, he exercises direct control over the environment represented in the simulation.

The model presents the resources used in base processes in terms of specific identifications the user provides. For example, the user may specify several aircraft types, several dozen types of maintenance personnel, perhaps a hundred different kinds of Aerospace Ground Equipment, and several hundred kinds of spare parts; the resource mix and the total resources included are constrained only by computer capacity. For each identified resource, the model keeps track of the amount available, amount in use, amount due-out, and so forth in accordance with resource interactions in accomplishing support requirements during the simulation.

While a simulation is in progress, the user may apply embedded decision routines. These determine whether specified performance goals are being met and, if not, selectively augment resource levels, according to a cost-effectiveness criterion, until the desired performance objectives are attained. The main output of the simulation model is a performance summary report produced at specified intervals during the simulation. This report presents summary statistics in six functional categories: operations, aircraft, personnel, shop repair, supply, and equipment. Consolidating model results in one report helps discern the interactions of the functional support areas in response to mission requirements. In addition to this report, the model provides, at user-option, separate status reports for resources, missions, jobs in process, and jobs back-ordered at specified simulation times.

Using data produced by the main simulation model, the postprocessor gives the user two kinds of products. One consists of the same data contained in the performance summary report but presented in graphical form. Thus it provides plots of fill rates, aircraft utilization, personnel utilization, and other summary results over simulated time, each on a separate page. The second product consists of graphical displays for selected airplanes. Each display identifies the processes incurred and the time-in-process for both the aircraft and components removed from the aircraft for shop repair. The production of either or both products of the postprocessor is optional in any given run.

For normal operation of the overall model, the preprocessor is first run to obtain a data base for the simulation. Then a number of simulation runs are made changing certain variables between runs as needed for particular study objectives. The postprocessor may or may not be exercised for each run, again depending upon the study objectives. Run results are then consolidated, compared, evaluated, and analyzed. For large studies, this mode of operation may be repeated several times, using a significantly different data base for each set of runs.

APPLICATION AREAS

The main application of the Logistics Composite Model is the analysis of alternative repair policies. The model is uniquely structured for such studies, since it can represent different kinds of repair processes by means of input data only rather than by changes in the computer programs. Thus L-COM can analyze any postulated repair policy, with corresponding variations in the repair processes involved.

Because of the flexibility built into the model for repair-level analyses, it may also be used for many studies concerning overall base operations; in particular, because it can explicitly represent many specific resources, it can perform studies involving resource allocation and utilization. For example, L-COM can analyze alternative policies for stocking reparable spares, allocating maintenance personnel across shifts, providing ground support equipment, and so forth. Using the model for these kinds of analyses allows the analyst to measure the overall effects upon mission performance as well as functional interactions.

Another application area is the determination of support requirements for alternative flying programs. First, the model may be used to measure the adequacy of a given resource mix to support different flying programs, either under steady-state conditions as for a peacetime training program or in a dynamic environment as for support of a contingency operation. This helps determine how support limitations affect flying programs. Second, the model may assist in determining a preferred mix of resources to support a given set of flying programs. For these analyses, the decision routines are particularly valuable since they identify limiting resources and provide guidance regarding how much augmentation they need to sustain an acceptable level of mission effectiveness.

Another major application area is the determination of logistics support requirements for airplanes being designed and the investigation of the logistical consequences of alternative designs for aircraft components and ground support equipment. Provided for this application are engineering estimates for reliability factors, repair processes involved, and other portions of the input data. The model can vary, from one component to another, the level of detail in which repair processes are represented. Thus, for particular components subject to design evaluation, greater detail can be included for more precise measurements concerning the utilization of maintenance personnel, AGE, spares, and other resources, as well as the effects upon overall weapon system performance.

With slight modifications, the L-COM model may be applied in a number of other areas. In principle, it can be applied wherever the environment can be represented as a task network with each task requiring time and/or resources of specified kinds. The kinds of processes that the task network structure can accommodate will be described in detail in Sec. II.

LANGUAGE

The computer programs for the Logistics Composite Model have been written in SIMSCRIPT,[*] which seems the most appropriate language for a simulation model of this size and scope. In particular, the dynamic storage allocation and initialization features of the language enable the use of the model for large problems. Other features of the language facilitate the structuring of the model and the programming effort.

Since this report is oriented toward an overall description of the model and the logical processes that it contains, references to particular terminology of the SIMSCRIPT programming language will be kept to a minimum. Where necessary, however, such terms will be defined in context with their use in the text.

CONTENTS

The foundation and main distinguishing feature of the Logistics Composite Model is the concept of representing the environment as task networks. Because of its importance, this concept is described in some detail in the next section. Following this, the model itself is described, with separate sections being devoted to the separate programs included in the overall model.

Section VII gives several operating characteristics of the model such as the size of problems it can accept and the computer time required to operate it. Section VIII discusses the validation requirements of the model and a brief summary of validation experience in context with the PACER SORT project. Finally, Sec. IX describes plans for future extensions and refinements of the model.

Two appendixes are included. One contains a list of data elements that the user provides to the model. The other contains a summary description of the major subroutines included in the main simulation model; similar descriptions for the pre- and postprocessors are omitted.

---

[*] H. M. Markowitz, B. Hausner, and H. W. Karr, *SIMSCRIPT: A Simulation Programming Language,* The RAND Corporation, RM-3310-PR, November 1962.

OTHER DOCUMENTATION

Several other documents pertaining to L-COM are being prepared or planned. Included are the following: Summary Report; User's Reference Manual; Programmer's Reference Manual.

The Summary Report consists of brief, overall descriptions of the model and its potential use in planning studies. The User's Reference Manual contains detailed specifications for developing input data, descriptions of logical processes included in the model, descriptions and examples of output reports, procedures for running the model on a computer, and general guidance concerning the use of the model in particular application areas. The Programmer's Reference Manual contains a detailed narrative description of the computer programs for the model, as well as the SIMSCRIPT program listings. These three documents are being prepared by personnel in the Simulation Center at Headquarters AFLC.

In addition to the above, it is anticipated that a series of reports describing the application and results of the model in context with particular studies will be produced. These will probably be developed by organizations and study groups using the model.

II. TASK NETWORKS

The main feature and unique characteristic of the Logistics Composite Model is the way the support environment can be represented in the model logic. This representation, consisting of task networks developed by the user, constitutes the foundation of the model and provides the flexibility needed to explore many problem areas. This section describes the task network concept in some detail; later sections show how the technique is used in the overall simulation.

GENERAL CONCEPT

The basic idea of the task network approach is that all processes in the environment can be expressed in network form, as illustrated in Fig. 2. In fact, the first thing an L-COM user must do is construct such a network for the processes he wishes to simulate.

The network consists of lines, representing separately identified tasks, and circles (or nodes), representing the start and completion of tasks. In general, tasks are defined as processes or operations that require, for their accomplishment, certain lengths of time and specified types and amounts of resources. Tasks that might be identified to represent aircraft support functions at a base are as follows:

Airplane flight;
Airplane refueling;
Weapons loading;
Flight-line check of a component;
Flight-line repair of a component;
Flight-line removal and replacement of a component;
Bench-check of a component in a repair shop;
Repair of a component in a repair shop;
Obtaining a spare part from base supply;
Transportation of spare part to the flight-line;
Replacing a spare part from depot supply.

These tasks, which only represent those the user may define, all require time and resources.

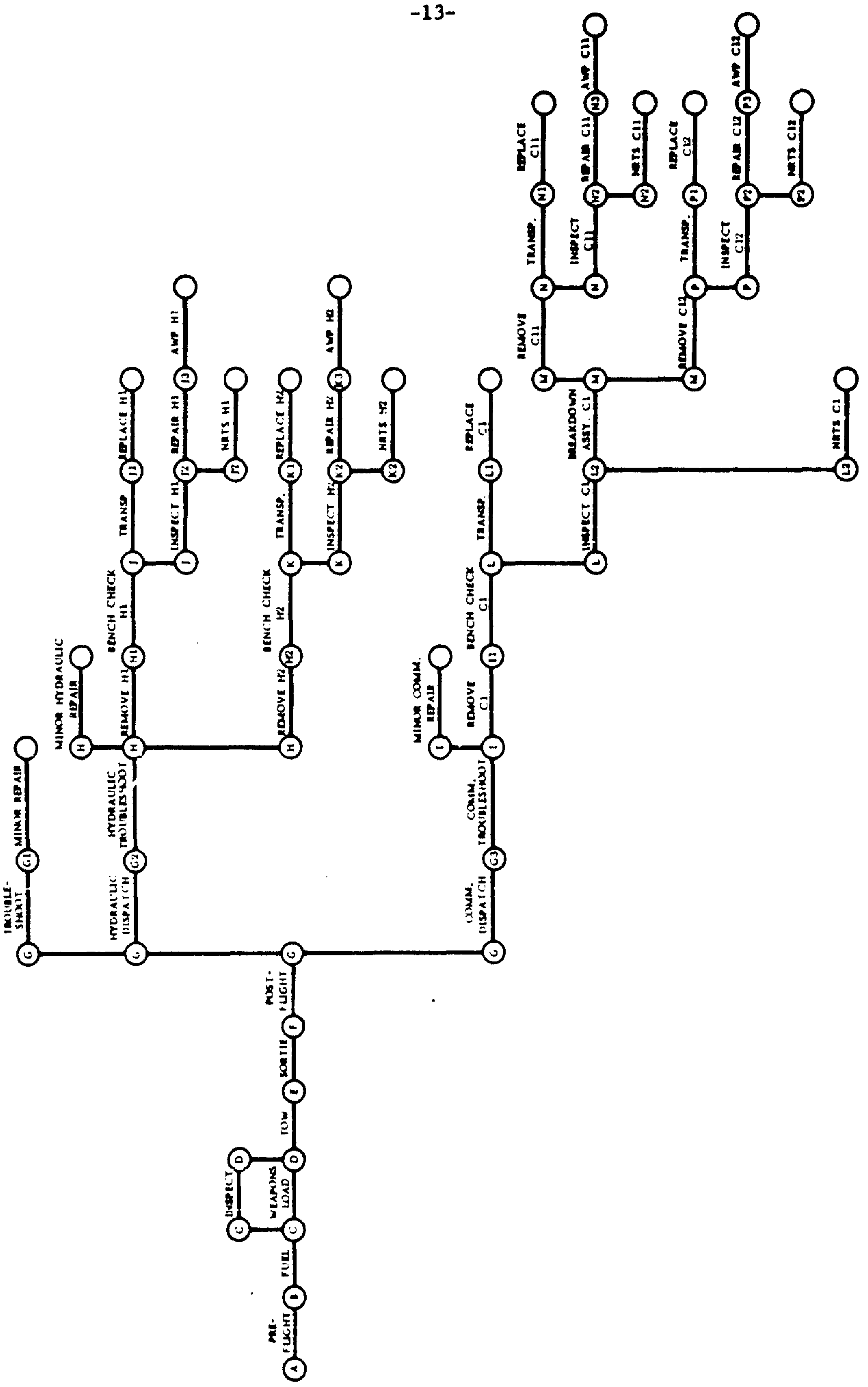


Fig. 2 -- Example task network

In addition to identifying all of the processes to be simulated, the network indicates how they interrelate in terms of sequence of accomplishment. As indicated in Fig. 2, some tasks must be performed in strict sequence; others may be performed simultaneously or in parallel. In some cases, a task cannot be started until a number of earlier parallel tasks have all been completed.

Even though the network identifies all possible tasks included in the simulation, not all of them will necessarily be incurred in a particular instance. Suppose, for example, that an airplane is exposed to the processes shown in Fig. 2 involving preflight tasks, the flight itself (considered as a task), and various postflight tasks. Some of these, such as refueling and the flight itself, must always be incurred. Others, such as unscheduled maintenance after the flight, may or may not be incurred; the selection is subject to chance or the probability of system failure during the flight. As described later, simple conventions have been established to identify which tasks will happen with certainty, and which will occur by chance as the simulation progresses.

During simulation, the task network is applied to a number of airplanes simultaneously. At a particular instant of time, one airplane may be in the fueling task, another may be in the flight task, others may be in various kinds of unscheduled maintenance tasks, and so forth. The simulation model keeps track of each airplane and where it is in the network. It also selects which tasks each airplane will undergo, according to occurrence probabilities based upon component reliability data and other factors.

Portions of the task network may represent shop repair processes for components removed from aircraft. For these components, the processing is similar to that for airplanes. The model keeps track of each component as a separate entity and selects appropriate tasks involved in the repair. During repair, assemblies may be removed from the component (as indicated by corresponding tasks in the network) and undergo separate processing. Again, the model tracks each assembly through its portion of the overall network. There is no logical limitation on the detail of component breakouts of this kind; the only limitations are the user's ability to describe the processes involved in network form and the computer's storage capacity.

In general, describing the environment to be simulated in task network form enables the user to specify any degree of detail or complexity that he finds necessary for his study objectives. An additional feature permits the network to be designed in sections, thereby simplifying and reducing the input data required. Thus a separate section which is utilized in more than one place in the total network can be defined only once and called from several locations within the network. An example of this might be maintenance actions applicable to both pre- and postflight phases of the airplane operation. There is no limit on the number or size of these sections. This feature, as well as other characteristics of the network formulation, are further described below.

TYPES OF TASKS

In developing a task network, each task must be assigned a unique name and all except the terminal nodes must be numbered or lettered in some fashion. The only restriction on how tasks and nodes are labeled is that they must be uniquely identified. In addition, each task must be assigned a code which the model uses in selecting the particular tasks for a given airplane, component, or assembly. Seven different codes have been established for this purpose, identified as follows:

Code	Definition
D	Do the task
S	Time constrained task
B	Constrained by prior task completions
A	Nonmutually exclusive selection
E	Mutually exclusive selection
F	Selection governed by failure mechanism
C	Call subnetwork

Each task is assigned only one code. Some codes require additional information. This information and the meanings of the codes are further described below, with portions of the network shown in Fig. 2 being extracted as examples.

Code D--Do the Task

Code D signifies that the task will always be selected each time it is encountered, and that its selection is not subject to chance or any other criteria. The code is usually assigned to sequential tasks as illustrated in Fig. 3. Processing will continue past a Code D task and into subsequent portions of the network.

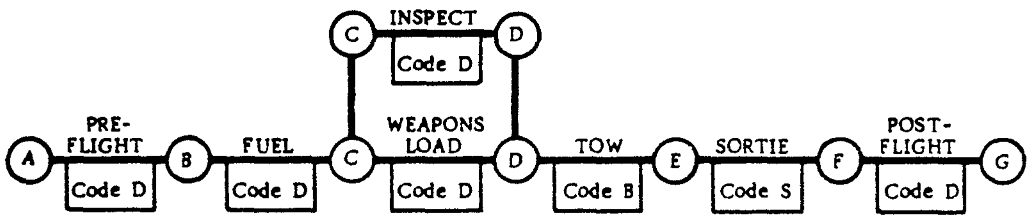


Fig. 3 -- Example of Code D, B, and S assignments

Code S--Time Constrained Task

This code is assigned to the sortie task, or any task whose starting time is to be controlled by the mission data. When this code is encountered on a task, mission data are referenced in order to establish the earliest possible starting time. If the preceding task is completed and other mission requirements are satisfied (to be covered in Sec. III), the task will be started. The assignment of this code is illustrated in Fig. 3.

Code B--Constrained by Prior Task Completions

A task is considered normally constrained if it cannot be started until a number of preceding tasks in parallel have all been completed. In the model, the assignment of Code B to a constrained task insures that all tasks leading directly to it are completed before attempting to start it. Failure to assign this code would allow the constrained task to be started as often as there are tasks leading to it, resulting in multiple processing of all following tasks. Figure 3 also illustrates the use of this code.

Code A--Nonmutually Exclusive Selection

This code is assigned to each task in a set of parallel tasks that may be selected independently. In the operation of the model, each parallel task is separately and independently checked to determine whether or not it will be selected. Selection is based upon a probability factor that must be assigned whenever Code A is used. This factor represents the fraction of times the task will be selected on an average. For example, if the assigned factor is 0.10, the task will be selected once out of every ten times it is encountered on an average. The model selects a task by generating a random number between zero and one, and comparing it against the probability factor. If the random number is less than or equal to the factor, the task is selected for processing; otherwise, it is not selected. In this way, the actual task selection is subject to chance.

As illustrated in Fig. 4, each of the parallel tasks may be assigned a different probability factor. Since selections are subject to chance, none, one, more than one, or all tasks in the set may be selected when encountered. Over the long run, however, as a number of aircraft or components are exposed to this portion of the network, the selection frequency will approximate the assigned probability factors.

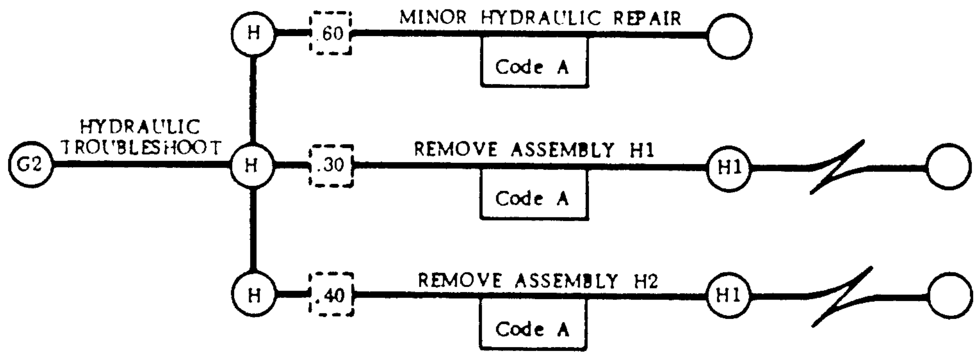


Fig. 4 -- Example of Code A assignment

Code A is often used to determine which parts cause the failure of a component or assembly. This code represents real-world uncertainties, since it is never known in advance which parts will cause a given assembly to fail. Only expectations, as represented by the probability factors described above, can be determined from a large number of previous assembly failures.

Code E--Mutually Exclusive Selection

Code E is assigned to each member of a set of parallel tasks wherein one and only one task is to be selected according to respective probability factors. The selection procedure is similar to that for Code A. Each task is assigned a probability factor, and a random number is again compared against the probability factors in the selection procedure. In this case, however, only one random number is used for the whole set of tasks and is compared against the sum of probability factors as successive members of the set are considered. The particular task for which the random number is first, less than, or equal to the cumulative value of the probability factors is selected for processing, and the procedure terminates since only one task is selected. The probability factors assigned to tasks in the set must add up to one to insure that a member of the set is selected.

The Code E assignment is illustrated in Fig. 5. Notice that the probability factors add up to one. This code is often used to determine whether a particular assembly should be repaired at base or depot level. The probability factor for depot repair is referred to as the NRTS (Not Reparable This Station) percentage. Code E enables simulation of this real-world process: whether a given assembly is repaired at base or at depot is subject to chance, but the choices for a large number of the assembly will be distributed as indicated by the NRTS percentage.

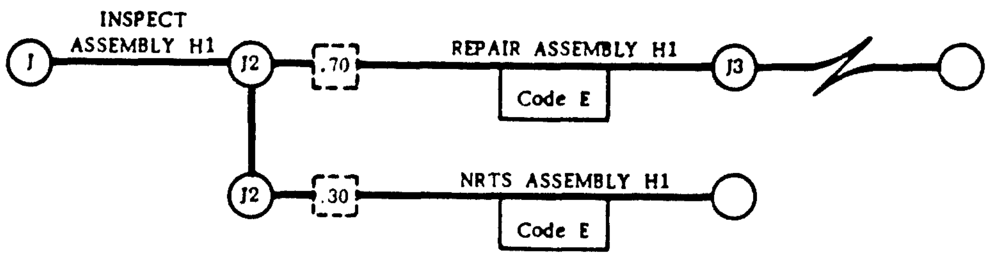


Fig. 5 -- Example of Code E assignment

Code F--Selection Governed by Failure Mechanism

As aircraft undergo flying operations, malfunctions of particular components may occur that must be corrected before the airplane is again available for use. These kinds of failures, as well as others, can be represented in the simulation model by a failure mechanism described later. As part of this logic, however, Code F must be assigned to certain tasks.

In principle, Code F is used to select those tasks leading to and involving the repair of an airplane or component that has had a lower level part failure. For a given exposure to the network, the failure mechanism determines *if* the tasks are to be selected. Code F determines *which* tasks are involved and subject to selection.

The assignment and use of Code F may best be described by referring to the example network in Fig. 2 and the extracted portion shown in Fig. 6. Suppose that when a particular airplane undergoes the simulated flight task, the hydraulic pump (Assembly H1) fails and must be replaced after the flight. In order to get to the tasks involving the removal, replacement, and repair of the pump, the task representing the dispatch of hydraulic system technicians must be selected. This task is assigned Code F. It tells the model that *when* the pump fails, the dispatch and subsequent tasks must be accomplished; and if the pump does not fail, these tasks are not selected. In this way, Code F controls the entry into lower levels of the task network in accordance with malfunction determinations.

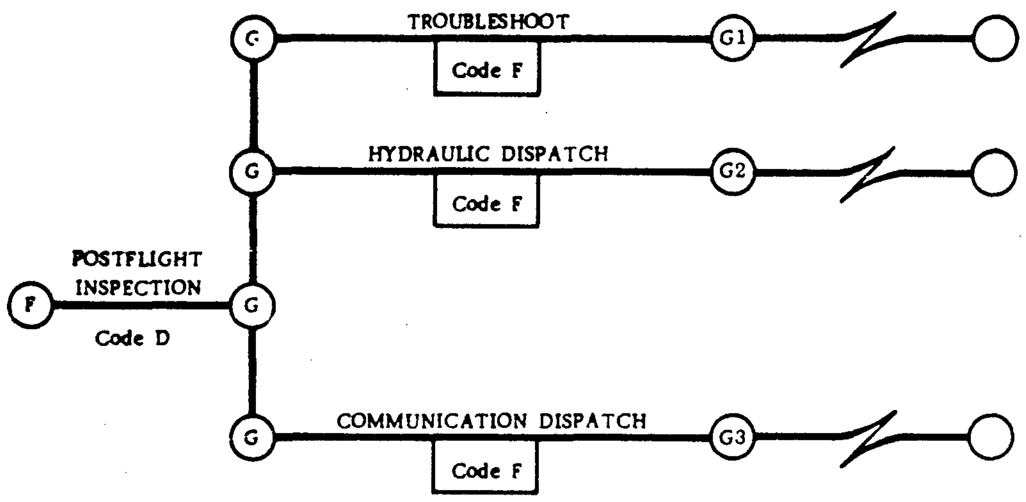


Fig. 6 -- Example of Code F assignment

Code C--Call Subnetwork

When a task with Code C is encountered, it is to be replaced by a separately constructed subnetwork of tasks. A Code C task is actually a "dummy" in the sense that it does not require time or resources but merely identifies the subnetwork to insert at this point. The first task in the subnetwork is processed, then the second, and so on. When all subnetwork tasks are completed, the processing reverts back to the main network and the task or tasks immediately following the Code C task are next considered. Code C is illustrated in Fig. 7.

A subnetwork of tasks may be constructed just like the main network itself. All of the task type codes may be used, including Code C. Thus, a main network may call a subnetwork which, in turn, may call a sub-subnetwork, and so forth. This feature enables an overall task network to be constructed in separate sections and allows a particular section to appear in several places without duplication. In more sophisticated uses, a subnetwork can even call itself as, for example, in simulating repeated attempts to repair an article where the success of each attempt is subject to chance.

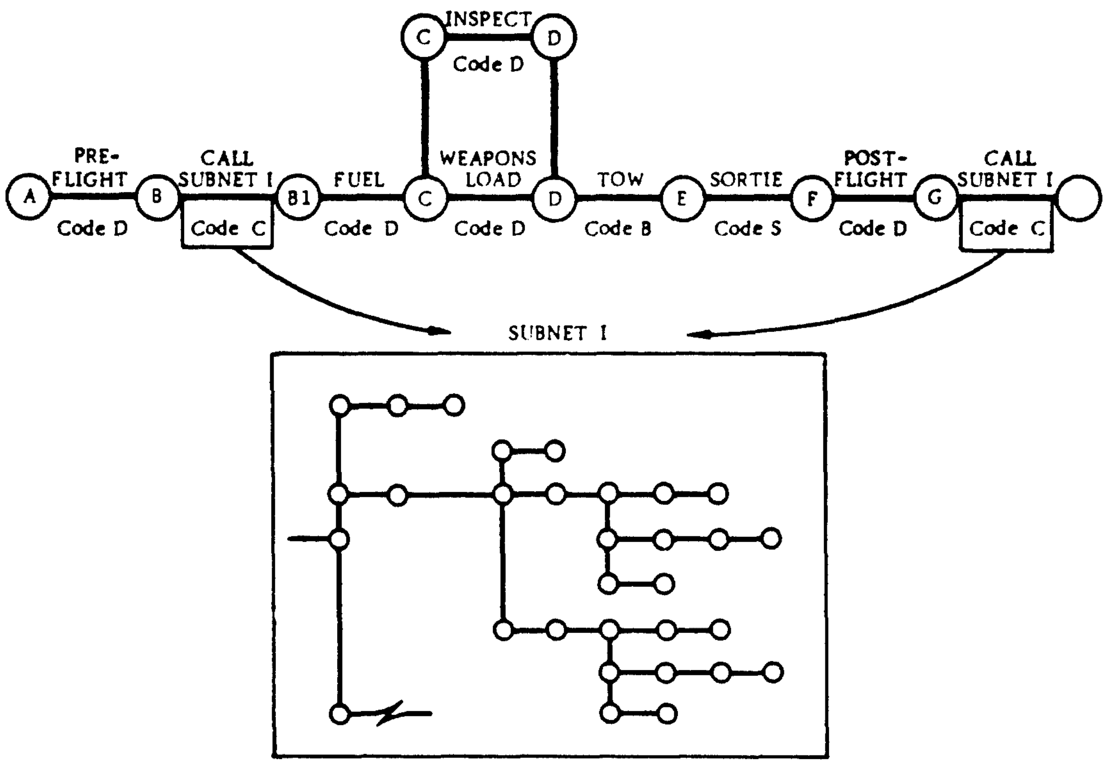


Fig. 7 -- Example of Code C assignment

NETWORK PROCESSING

The task network described above controls which processes will be selected and represented in the simulation. Normally, entry into the network is caused by a mission requirement. For each airplane the mission requires, a record is established that represents the aircraft through subsequent operations. According to type of sortie involved in the mission, an initial or starting task in the network is selected, and the next task is identified in a data element of the airplane record. When the first task is completed, the data element determines where to go next in the task network. When a new task is selected, its successor task is identified in the airplane record. In this way, the airplane's location in the task network at any time is known. The codes assigned to the various tasks determine whether or not they will be incurred as the airplane progresses through the network during the simulation.

When the network indicates the removal of a component from the airplane, the model establishes a new record to represent the component. This record is processed through its portion of the network in a fashion similar to that for the airplane. If an assembly is removed from the component, the model creates and processes a record for the assembly. This procedure continues to the extent prescribed by the task network.

As tasks are processed, resource requirements are determined. The availability and interaction of the resources determine whether the task will start right away or be delayed. The logic involved in this decision is described later. Whenever a task is completed, control reverts back to the network to determine what should be considered next. This procedure continues until the airplane, component, or assembly returns to a serviceable or ready-for-use status.

INPUT DATA FOR TASK NETWORKS

After the task network is constructed and annotated with task and node labels, codes, and selection probabilities, the data are transcribed onto an input form. Such a form, partially completed with data from the example network in Fig. 2, is shown in Fig. 8. Each task is represented by a line of data. The first field contains a preprinted card number to indicate the kind of data being input to the computer. The "Prior Node" field contains the label for the beginning node of the task. The "Selection Mode" field contains the task type code as previously described; if no entry is made in this field, the computer will automatically insert a Code D--DO THE TASK. The "Task I.D." field contains the label for the task itself. The "Next Node" field contains the label for the ending node of the task. The "Selection Parameter" field contains the selection probability factors; this field must be completed whenever a task type Code A or E is assigned. Finally, the "Task Description" field may be used to enter any desired information further describing the task for reference purposes; the model does not use this information in any of its processing.

As can be seen, preparing the input data for task networks is quite simple and straightforward once the networks have been constructed.

LOGISTICS COMPOSITE MODEL

SECTION II - SIMULATION MODEL INPUTS
FORM 11 - TASK NETWORK

CARD I.D.	PRIOR NODE	TASK I.D.	NEXT NODE	SEL. MODE ●	SELECTION PARAMETER	TASK DESCRIPTION
11	A	PREFL	B	D		PREFLIGHT
	B	FUEL	C	D		FUELING
	C	INSPEC	D	D		INSPECTIØN
	C	W-LØAD	D	D		WEAPØNS LØAD
	D	TØW	E	B		TØW
	E	SØRTIE	F	S		SØRTIE
	F	PØSTFL	G	D		PØSTFLIGHT
	G	TS-ØM	G1	F		TRØUBLESHØØT
	G	HYDDIS	G2	F		HYDRAULIC DISPATCH
	G	CØMDIS	G3	F		CØMMUNICATIØNS DISPATCH
	G1	REPAIR		D		MINØR REPAIR
	G2	TS-HYD	H	D		TRØUBLESHØØT-HYDRAULIC
	H	HYDREP		A	.60	MINØR HYDRAULIC REPAIR
	H	REM H1	H1	A	.30	REMØVE ASSEMBLY H1
	H	REM H2	H2	A	.40	REMØVE ASSEMBLY H2
	H1	BCH H1	J	D		BENCH CHECK ASSY H1
	J	TRANH1	J1	D		TRANSPØRT ASSY H1
	J1	REPLH1		D		REPLACE ASSY H1
	J	INSPH1	J2	D		INSPECT ASSY H1
	J2	REP H1	J3	E	.70	REPAIR ASSY H1
	J2	NRTSH1		E	.30	NRTS ASSY H1
	J3	AWP H1		D		AWAITING PARTS FØR H1
	H2	BCH H2	K	D		BENCH CHECK ASSY H2
	K	TRANH2	K1	D		TRANSPØRT ASSY H2
	K1	REPLH2		D		REPLACE ASSY H2
	K	INSPH2	K2	D		INSPECT ASSY H2
	K2	REP H2	K3	E	.90	REPAIR ASSY H2
	K2	NRTSH2		E	.10	NRTS ASSY H2
	K3	AWP H2		D		AWAITING PARTS FØR H2
	G3	TS-CØM	I	D		TRØUBLESHØØT-CØMMUNICATIØNS
	I	CØMREP		A	.50	MINØR CØMMUNICATIØNS REPAIR
	I	REM C1	I1	A	.10	REMØVE ASSY C1
	I1	BCH C1	L	D		BENCH CHECK ASSY C1
	L	TRANC1	L1	D		TRANSPØRT ASSY C1
	L1	REPLC1		D		REPLACE ASSY C1

● SELECT MODE: A = Any B = Constrained, C = Call sub-network D = Do task E = Either F = Failure, S = Time constrained

Fig. 8 -- Example of data input form

However, the initial drawing of the networks to capture the environment adequately can be challenging, since it requires not only technical familiarity with the processes being simulated, but also judgment with respect to the level of detail appropriate to particular study objectives. Since the task networks constitute the foundation of the L-COM model, careful attention is required for their construction. But if properly developed and applied, the task network concept places in the user's hands the ability to describe the environment as he pleases, without being forced to accept a particular description that would otherwise be frozen into the model logic. With experience in its use, this distinguishing feature can be extremely valuable in applying simulation to many problem areas and planning studies relating to base operations.

III. PREPROCESSOR

Any simulation model of reasonable size and scope requires a considerable amount of input data. As more features of the real-world environment are included, more data are required to describe them. To help make this data preparation problem as easy as possible for the user, it is desirable that he have input forms explicitly designed for his convenience, rather than for the simulation model itself. This, however, requires a computer program to translate the data into the particular formats the simulation program uses. The Logistics Composite Model has a preprocessor, or input program, developed for this general purpose. The program is described in this section.

FUNCTIONS OF THE PREPROCESSOR

In the L-COM model, the preprocessing routine accomplishes the following three functions:

Edits and error-checks input data.

Reformats and reorganizes input data for entry into the simulation.

Develops sortie requirements according to a specified flying program.

In the edit and error-checking function, the preprocessor analyzes data elements to insure their completeness and accuracy insofar as possible. For example, codes such as the task type described in Sec. II are inspected for illegal entries. Also, certain consistency checks are made; for example, the data describing task networks are analyzed to insure that all the tasks are properly connected. Whenever errors or inconsistencies are detected, messages are printed out for user attention and correction.

As part of the edit process, data field descriptions designed for user convenience are translated into those required by the simulation model. For example, a user may insert the number five in a data field as either "5" or "5." or "5.0", and the number may be placed anywhere within the designated field of the input form. The preprocessor will then convert the number, however entered, into a standard format. Similar features are provided for alphanumeric data and for time values that may be input by the user in days, hours, minutes, or combinations thereof, according to his convenience.

After the data are edited, they are reorganized into formats required for input to the main simulation model. These formats are prescribed by the SIMSCRIPT language in which the model is written. The reformatted data are identified as "initialization" data for the simulation model.

On one of the input forms, the user may specify a flying program to include in the simulation. The form provides several options for specifying the flying program, so the preprocessor can generate specific mission requirements in terms of various types of sorties to be accomplished during simulation. The corresponding data are produced in formats prescribed by the SIMSCRIPT language. These are referred to as "exogenous event" data, since they represent real-world events coming from outside of the support environment being simulated.

The preprocessor also produces reports containing the results of these functions. In particular, it provides listings of the input data for reference purposes.

INPUT FORMS

To provide data required by the main simulation model, the preprocessor uses the following input forms:

Form Number	Description
10	Performance Summary Report Specifications
11	Task Network
12	Task Definitions
13	Resource Definitions
14	Failure Clock Decrements
15	Distributions
16	Shift Change Policies
17	Mission Entry Points
18	Priority Specifications
20	Sortie Generation Data

These forms, illustrated in Fig. 9, are briefly described below with

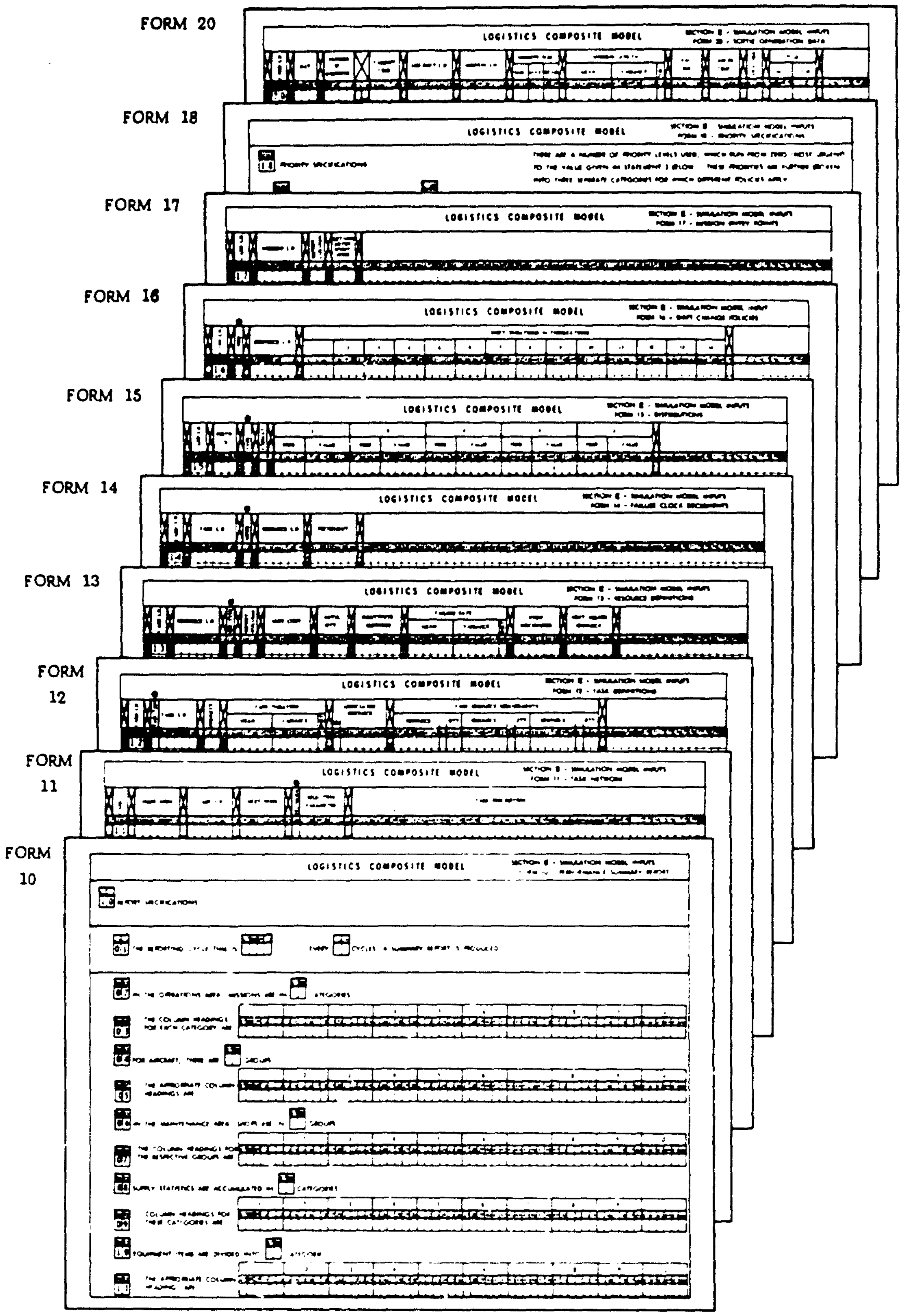

FORM 20
FORM 18
FORM 17
FORM 16
FORM 15
FORM 14
FORM 13
FORM 12
FORM 11
FORM 10
LOGISTICS COMPOSITE MODEL

Fig. 9 -- Input forms used by L-COM Model

respect to the kinds of data they contain; specific data elements contained in these forms are listed in Appendix A.

Form 10--Performance Summary Report Specifications

On Form 10, the user specifies the frequency with which the performance summary report is to appear. A basic reporting cycle of a day, week, or other period may be prescribed, with an overall summary being produced weekly, monthly, or some other multiple of the basic cycle. In addition, entries on this form enable the user to specify what should be printed as columnar headings on the performance summary report. In general, the input data give the user control over the meanings of the columnar breakouts of this output report.

Form 11--Task Network

Form 11 contains data for the task networks. It is described in Sec. II.

Form 12--Task Definitions

Form 12 provides additional data for each different kind of task. Included are task priority, task duration parameters, and task resource requirements in terms of kinds and amounts of resources needed to accomplish the task.

Form 13--Resource Definitions

For each resource required by one or more task(s), additional data are provided on Form 13. Included are identifications of substitutes, if any, unit costs, authorized quantities, failure parameters, and a resource type code identifying resources as aircraft, parts, men, AGE, or facilities.

Form 14--Failure Clock Decrements

When certain tasks occur, a number of particular resources may be exposed to possible failure. Form 14 identifies these tasks and

BEST AVAILABLE COPY

associated resources, and gives the kind and/or amount of exposure to failure. These data are used in the model's failure mechanism.

Form 15--Distributions

Several factors in the simulation, such as task durations and times between part failures, are subject to uncertainty, which is represented in the model by drawing random values from probability distributions. If the user wishes to use actual data, as obtained from real-world observations, Form 15 is used to enter probability distributions in terms of histograms. Otherwise, he may specify any one of several standard distributions such as the normal or exponential functions that the model contains.

Form 16--Shift Change Policies

Any resource included in the simulation may be subject to a shift change policy whereby its authorized level is changed over time. Form 16 contains data specifying the resources subject to shift changes, the time durations of the various shifts, and the authorized levels for each shift.

Form 17--Mission Entry Points

When a mission requirement occurs, available aircraft are selected for processing through the task network. Since the tasks involved may differ by type of mission, different initial entry points into the overall task network may be necessary. Form 17 identifies these initial points of entry.

Form 18--Priority Specifications

This form contains priority level definitions and other factors representing maintenance policies for expedited repair, preemption of in-process tasks, the use of overtime, and other policies as described in Sec. IV.

Form 20--Sortie Generation Data

Form 20 contains data prescribing the flying program. These data and the way the preprocessor uses them to generate sorties are further described below.

SORTIE GENERATOR

The following data characterize each mission requirement:

Type of mission.
Type of aircraft required.
Desired takeoff time.
Preparation time prior to takeoff (lead time).
Mission length.
Mission cancellation time.

Possible types of missions are: close support, interdiction, reconnaissance, training, and so forth; different types of missions may involve different kinds of support tasks, such as weapons loading and other preflight operations. It is assumed that all airplanes in a given mission are the same type. If different types are required, separate missions are specified.

The number of airplanes required for each mission is specified in terms of a minimum, a maximum, and spares. During the simulation, the maximum number will be flown if possible. If less than the minimum are available, the entire mission will be canceled. Spare airplanes, if any, are prepared for flight and held in reserve in case other mission airplanes cannot be made ready in time or would otherwise abort the mission.

The desired takeoff time occurs when all aircraft in the mission are ready to depart. If some or all of the airplanes are not ready, takeoff time is deferred, but no later than the mission cancellation time. If enough airplanes are not ready by this time, the mission is canceled.

The mission lead time represents the time prior to takeoff allowed for accomplishing presortie tasks, such as preflight maintenance or weapons loading. If, during the simulation, the actual presortie tasks extend beyond the scheduled takeoff time, the mission will be delayed or possibly canceled. If the airplanes take off, the mission length is the actual time spent on the sortie; it is the same for all airplanes in the mission.

One of the preprocessor's functions is to generate a list of missions with specific values for the above data elements. The user provides input data for this purpose on Form 20--Sortie Generation Data. On this form, several options are available with respect to the level of detail in which the mission data are provided. As one option, specific values may be assigned to the data elements for each mission. This is particularly useful for representing real world missions in the simulation. In the other options, specific values for one or more of the data elements are randomly selected from probability distributions or histograms, rather than being explicitly given by the user.

As an example, the number of missions of a given type to be accomplished during a specified day may be obtained (by the preprocessor) from a probability distribution. Similarly, takeoff times may be randomly determined from histograms representing mission frequencies as a function of time of day. The mission length may also be randomly determined from a probability distribution; if a standard distribution is used, the user normally supplies a mean and standard deviation. Finally, missions may be rescheduled at a daily interval over a specified time span. This feature enables the user to represent repetitions of the daily flying program over longer time intervals without repeating the input data.

The form for entering flying program data allows for any mix of the above options. Thus, some missions may be explicitly prescribed while others may be specified by the random processes described above. Certain mission types, such as training sorties, may be caused to occur repetitively throughout the simulation according to a prescribed pattern. According to the kind of data provided on the form, the preprocessor accomplishes the functions necessary to generate a list of specific missions for input to the simulation.

OUTPUT REPORTS

As part of its operation, the preprocessor produces the following kinds of reports:

Error Messages
Task Network Analyses
Mission Summary
Dictionary of Assigned Names
Initialization Data

For the error messages, relevant information is either separately printed or interspersed in the other reports according to type of error involved. These messages contain explicit information pertaining to and describing the error involved. The outputs of the task network analyses provide another class of potential errors. In addition to a printout of the task network data, results of various consistency checks are produced. Included are identifications of network entry points, undefined nodes, and nodes that are multiply-referenced in the network data. This information may be used to locate errors in the construction of the task networks.

The mission summary report contains results of the mission generation procedure. It lists all data pertaining to the missions generated by day, mission type, and aircraft type.

Printouts for initialization data are presented in two formats. One corresponds closely to the input forms the user fills out. The other represents the translation of this data for input to the simulation program; this format is prescribed by the SIMSCRIPT initialization procedure.

Aside from the error messages, the preprocessor outputs are used mostly for reference purposes. They contain, in a consolidated printout, all of the input data used by the mode. The outputs also include a summary of the total number of tasks, network nodes, resources, and failure clocks that are contained in the simulation.

The dictionary produced by the preprocessor references resource numbers and task numbers used by the simulation program to the respective names assigned in the input data. These references are required to relate certain outputs of the simulation model to the input data; they are also needed to produce postprocessor outputs.

IV. SIMULATION PROGRAM

The simulation program constitutes the main part of the Logistics Composite Model. This program represents and simulates interactions of base operations and support processes in accomplishing mission requirements. This section describes the structure and logical procedures of this program.

GENERAL DESCRIPTION

The main simulation model, using the task network concept described in Sec. II and responding to mission requirements generated as described in Sec. III, is designed to simulate a broad range of aircraft operations, scheduling, maintenance, and supply functions at an Air Force base. For each mission requirement, the program internally controls the processing of each aircraft toward the completion of the mission as illustrated schematically in Fig. 10.

As in real life, the model attempts to complete all missions, but some cannot be completed as scheduled. Actions which could cancel a mission are the unavailability of sufficient aircraft to schedule into presortie processes or the loss of aircraft to unscheduled maintenance according to failures detected in presortie tasks.

After the mission is completed, the airplanes undergo postsortie maintenance as prescribed by the task network. Normally included are debriefing and troubleshooting tasks wherein aircraft system failures are detected and their correction begun. This phase of the simulation generates demands for spare parts from supply, and simultaneously generates reparable assemblies for maintenance in one or more of the base repair shops or, in the case of NRTS items, depot-level repair facilities.

At this point, the maintenance simulation follows two distinct paths, one for the aircraft itself (flight-line maintenance) and one for the reparable assembly (shop maintenance). Flight-line maintenance consists of those tasks required to troubleshoot and remedy failures on the aircraft or to replace faulty items with serviceable spares.

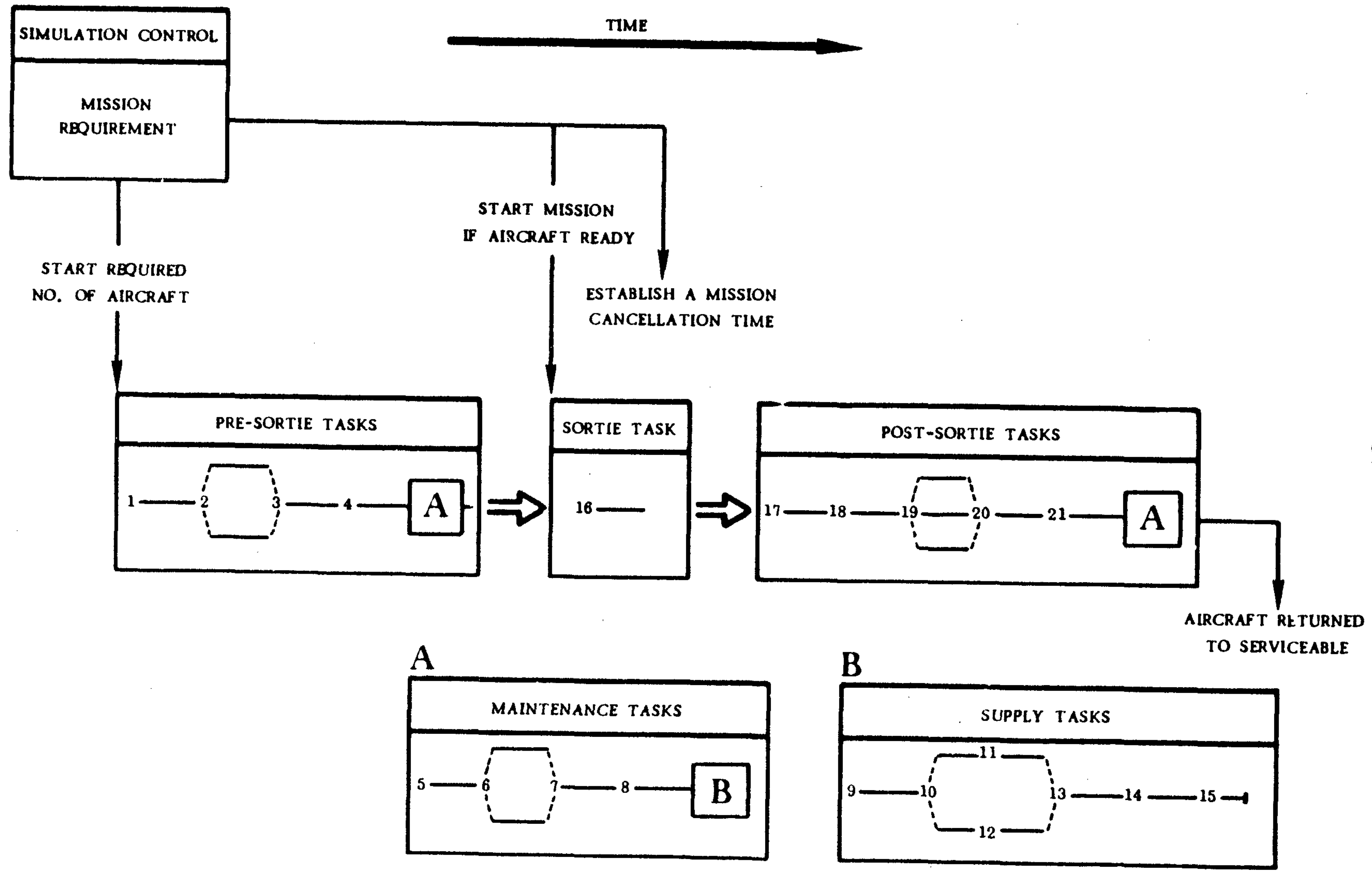


Fig. 10 -- Schematic of main simulation model

After all aircraft maintenance is completed, the airplane is returned to a serviceable pool.

Shop maintenance tasks are those base or depot processes required to diagnose faults in the reparable assembly and remedy failures by repair and/or replacement of one or more of its subassemblies from serviceable stock. In this process, a second indenture reparable may be generated for further base or depot repair. As prescribed by the task network, this procedure can be repeated for successively lower levels of indenturing, with each reparable being followed through its set of maintenance tasks.

Whether to repair an item at base or depot is a random choice based upon the input NRTS percentage for the item. In each case, the reparable follows a specific set of tasks; for the depot, the processes may be abstracted into a single task representing the overall depot order and shipping time. Upon completion of designated tasks, the unit is returned to supply as a serviceable item and appropriate stock levels are incremented.

In simulating repair and other kinds of tasks, conflicts may arise over resource availability. There may be insufficient amounts of a given resource item to satisfy all requirements at a particular time. To resolve these conflicts, a priority system is used as well as logical processes for using substitute items, expediting parts out of repair shops, preempting lower priority tasks, and cannibalizing parts off of airplanes down for other reasons.

To simulate parts' failure which, in turn, controls the occurrence of repair tasks, the program logic provides a failure mechanism that uses failure probability distributions based upon parts reliability data to determine when failures occur. It also exercises appropriate portions of the task network.

A shift mechanism within the simulation program permits identified resources, such as personnel, to be utilized on a shift basis. Such resources are designated as working under specific shift policies; several policies may operate at the same time. At appropriate shift intervals during the simulation, levels of associated resources are incremented or decremented as required.

Data are collected during the entire operation of the simulation program and at specified times are produced in output formats. The main output is a performance summary report containing summary statistics for all major functional areas of the base operation. The user may also produce several status reports if he wants.

INPUT DATA

As illustrated in Fig. 1, the main simulation program receives four types of input data as follows:

Mission requirements data
Initialization data
Run specification data
Decision model input data

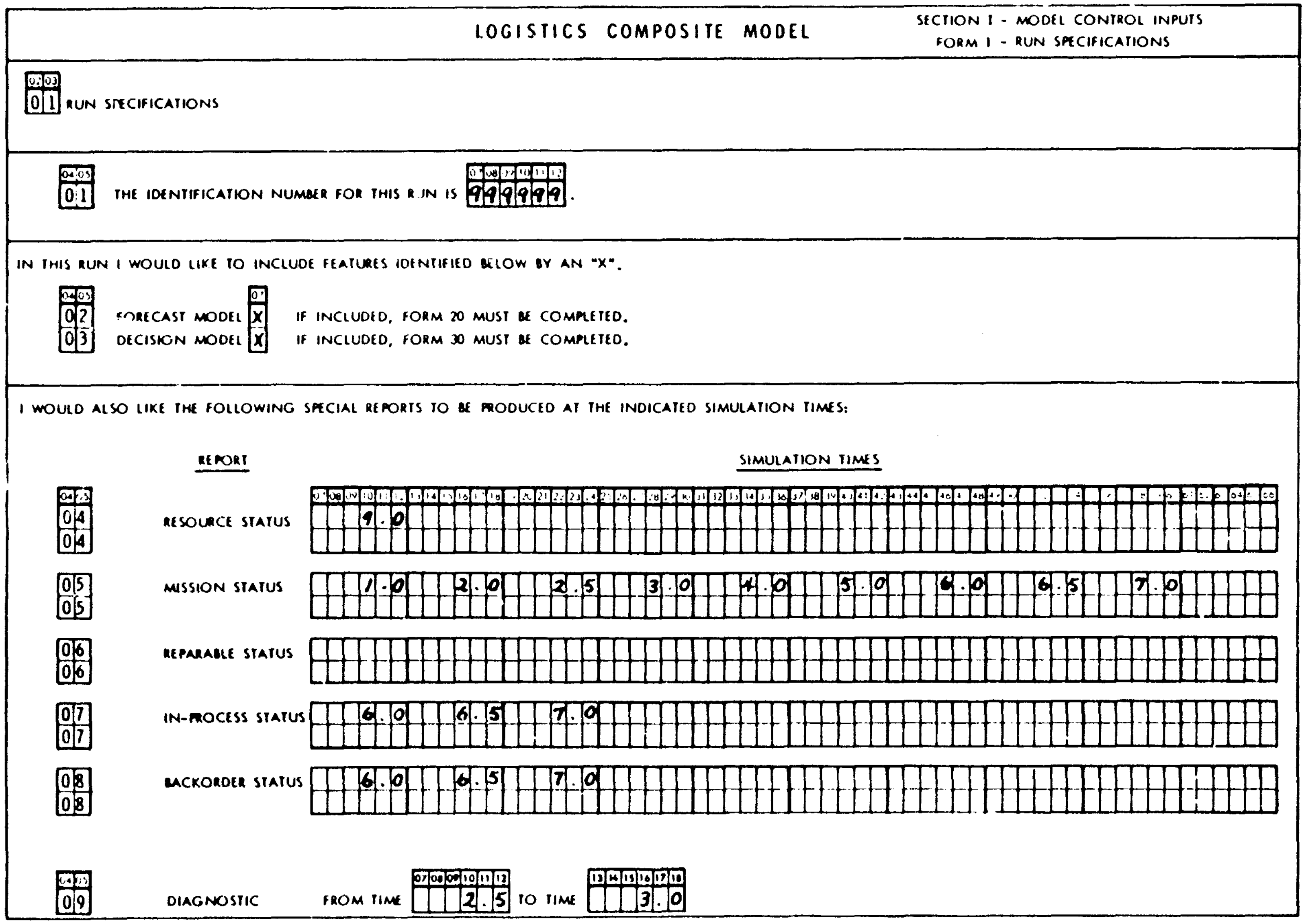
The mission requirements data, generated by the preprocessor for input to the simulation, specify the kinds of missions to be simulated, when they are to occur, what types and how many aircraft will be needed, and other factors as described in Sec. III.

The initialization data, prepared by the preprocessor for use by the simulation program, contain information in the following functional groupings or tables:

Control Table
Task Table
Resource Table
Shift Tables
Failure Data
Network Initial Entry Table
Report Data
Priority and Other Factors

Within each of these groupings, a number of data elements are provided by input forms as summarized in Sec. III.

In addition to data obtained from the preprocessor, a special form is used to specify particular features of the model that are to be exercised when the simulation program is run. As illustrated in Fig. 11, this form (Form 1--Run Specifications) permits the entry of a run identification number which is printed on all output reports, specifications about whether the forecast or decision procedures are



LOGISTICS COMPOSITE MODEL

SECTION I - MODEL CONTROL INPUTS
FORM I - RUN SPECIFICATIONS

01 RUN SPECIFICATIONS

01 THE IDENTIFICATION NUMBER FOR THIS RUN IS 999999.

IN THIS RUN I WOULD LIKE TO INCLUDE FEATURES IDENTIFIED BELOW BY AN "X".

02	FORECAST MODEL	X	IF INCLUDED, FORM 20 MUST BE COMPLETED.
03	DECISION MODEL	X	IF INCLUDED, FORM 30 MUST BE COMPLETED.

I WOULD ALSO LIKE THE FOLLOWING SPECIAL REPORTS TO BE PRODUCED AT THE INDICATED SIMULATION TIMES:

	REPORT	SIMULATION TIMES								
04	RESOURCE STATUS	9.0								
05	MISSION STATUS	1.0	2.0	2.5	3.0	4.0	5.0	6.0	6.5	7.0
06	REPARABLE STATUS									
07	IN-PROCESS STATUS	6.0	6.5	7.0						
08	BACKORDER STATUS	6.0	6.5	7.0						

09 DIAGNOSTIC FROM TIME 2.5 TO TIME 3.0

Fig. 11 -- Run specifications form

to be exercised, and identifications of special reports to be produced during the run. This form may be separately completed for each run.

If the embedded decision model is to be exercised during the simulation (as specified on Form 1), additional data must be provided as inputs to the simulation program. This data and corresponding forms will be described in the next section.

MISSION REQUIREMENTS

From a logical processing point of view, the simulation program starts when a mission requirement arrives at a particular point in simulation time. Aircraft availability data in the Resource Table are checked to determine whether sufficient airplanes are available to fly the mission. According to availability, the number of airplanes scheduled for the mission will lie between the minimum number required and the maximum number plus spares, as specified in the mission data. If the minimum number is not available, the mission requirement is set aside until airplanes are available from previous missions.

Each airplane assigned to the mission is started through preflight tasks according to the task network as represented by the Control Table. The first task to be incurred depends upon the mission type and is so identified in the Initial Entry Table. The task is started if the prescribed resources for the first task are available or are made available by the task processing described below. When the task is completed at a later point in time, reference is made to the Control Table to determine what task or tasks are next involved in the preflight operations. Again, the selected tasks are started if possible, and the procedure continues until all preflight tasks for the airplane are accomplished. At this time, the airplane is flagged as mission ready.

When mission time arrives, the mission is started if enough aircraft are ready. If not, it is delayed until either sufficient airplanes become available or the time to cancel the mission arrives, whichever is first. The airplanes being prepared for a mission that subsequently does not take place are returned to the on-hand pool after the tasks they are in at cancellation time are completed. Similarly, if the mission is canceled, all airplanes are returned to the on-hand pool in lieu of starting the sortie task.

FAILURE MECHANISM

When the flight task and other specially identified tasks are encountered, the model logic that causes item and equipment failures to be represented is called into operation. The tasks so identified are those that tend to cause failures in the real world because they involve the operation or stressing of airplanes or other equipment.

As part of the input data, resource items likely to fail are identified for each failure-causing task. Each such resource is viewed as owning a clock which tells, at any given time, how long it will be before its next failure occurs. Each time the associated task occurs, the clock is reduced by an appropriate amount, resulting in a shorter remaining time-to-failure.

The meaning of the clock and the amount by which it is decremented each time may be different from one type of resource to another. For example, failures for landing gears are related to number of landings and, therefore, its clock tells how many more landings will occur before a failure results. In this case, the clock is reduced by one (representing one landing) each time a flight task occurs. As another example, the airplane navigation system may operate whenever the airplane flies, and its failures, therefore, are related to cumulative flying hours. In this case, the clock for the navigation system is read as flying hours until next failure; it is decremented by the sortie length each time a flight task is incurred. Similarly, clocks for other kinds of resources may be scaled in terms of equipment operating time, calendar time, or other measures, and appropriately reduced when tasks that probably cause the failures are encountered.

Two actions are taken for each resource whose clock reaches zero. First, the clock is reset to some value selected from a failure probability distribution for the item. This value is normally a random draw from a distribution of times (flying hours, number of landings, etc.) between failures as determined from previous experience. Resetting the clock represents the uncertainty involved with respect to when the next failure will happen during the simulation.

The second action taken is to open up portions of the task network representing correction of the malfunction. As part of the failure data, all tasks whose sequence must be followed in effecting the repair of a given part are identified by a Code F in the task network. The last task in this sequence will have, as an additional identifier, the name of the resource whose clock determines the failure. This identifier is entered in the failure parameter column of the Network Data Form (Form 11). In this way, the necessary corrective actions, as identified in the task network, are taken whenever an item failure occurs.

In summary, the failure mechanism of the model simulates real-world failures with respect to what causes them, the frequency and randomness of occurrence, and the identification of tasks involved in their correction.

TASK PRIORITIES

In the processing described below, conflicts among tasks may arise because of a shortage of required resources. To resolve these conflicts, a priority system has been established as part of the model logic.

As one of the data elements in the Task Table, a priority may be assigned to each task in the network. Any convenient priority scale may be adopted, such as 1 to 10, 1 to 30, or 1 to 100. The tasks with the lowest priority numbers are considered the most important; for example, if two tasks are competing for the same resource, the one with the lowest priority number will obtain it.

For some rules used in task processing, the priorities may be categorized in three groups. If, for example, a priority scale of 1 to 30 is adopted, the respective groups might consist of priorities 1-10, 11-20, and 21-30. An even distribution, as in this example, however, is not necessary; the user may divide the priority scale into three contiguous regions in any way he pleases. These priority groupings are described below.

In addition to being assigned to tasks, the priority scale may be applied to missions. In this case, the priorities are used to determine which missions will obtain airplanes if there are not enough for all missions at a particular time. In general, mission priorities are used in the same way as task priorities, with airplanes being the applicable resource. As in the real world, the simulation program uses the priority system to represent the relative importance of tasks or missions when competing for s arce resources.

TASK START PROCEDURES

Whenever a task is selected for processing, as determined by the network representation in the Control Table, a common set of procedures are applied that either start the task or place it in a back-order status awaiting the availability of needed resources. The first step in this procedure is to determine, from data contained in the Task Table, what resources and how much of each resource are involved in accomplishing the task. It is further determined, from Task Table data, whether each resource is to be temporarily used by the task, consumed during the accomplishment of the task, or generated by the task. These possibilities are further described as follows:

> *Temporary Use.* The indicated resource amounts are used during the entire task, and are then returned for use by other tasks. This is normally applicable to such resource types as men, equipment, and facilities.
>
> *Consumption.* The resource is either used up by the task or it becomes an integral part of an airplane or a higher assembly; in both cases, the resource is not returned when a task is completed and is not available for any further use. This normally applies to consumable resources such as fuel and ammunition, and to repair parts that are installed in airplanes or components during the task.
>
> *Generate.* The resource is generated or created as a separate entity during the accomplishment of the task. This normally pertains to components or assemblies that are removed from airplanes or higher order assemblies and which undergo subse ient repair or other processes

as independent units. This feature can also be used to simulate work done that is not related to any specific resource.

In addition to resources involved in the task, as identified by Task Table data, there is always a specific resource item associated with the task. This item is the airplane, component, or assembly that is separately tracked through the task network and for which a data record exists to represent the item as it incurs appropriate tasks. Each task selected for processing is identified by primary item and associated data record.

After associated resources and resource requirements have been determined, it is necessary to determine whether resources to be temporarily used or consumed are available. For each such resource, the on-hand balance in the Resource Table is checked. If sufficient amounts of all needed resources are available, the on-hand balances are appropriately reduced, due-in balances (also data elements in the Resource Table) are correspondingly increased, and the task is started. If sufficient amounts are not available for one or more required resources, the task is processed further. This processing consists of a series of steps directed toward obtaining the required resources by other means as follows:

Use of substitutes
Expedited repair
Task preemption

Accomplishing these steps constitutes a filtering process in the sense that they are done in sequence and the search terminates at the step in which the needed resources are first found. The logic involved in each step is further described as follows:

Use of Substitutes

If there are insufficient amounts of certain resource items, other items in the simulation may be substituted. Such substitutes are identified in the Resource Table. Only one substitute may be listed for a given item; however, two items may list each other as substitutes.

When there is not enough of the prime item to meet a task requirement, as much of the substitute item as is available or needed to complete the requirement is allocated to the task. If prime and substitute items meet all task requirements, balances are appropriately adjusted and the task is started. Otherwise, the next step in the search for resources is taken.

Expedited Repair

If prime or substitute items cannot satisfy task requirements, other tasks currently using the needed resources are inspected to determine when they will be completed, thereby releasing the needed resources. If an in-process task will finish soon enough, the incoming task is set aside until the in-process task is completed. If not, an assumption for expediting completion of the in-process task is tested. This representation of expedited repair is accomplished by multiplying the remaining task duration by a specified fraction, such as 0.80. If the new completion time for the in-process task meets the waiting time criteria for the incoming task, corresponding actions are taken to expedite the former and set aside the latter.

For a given incoming task, several in-process tasks may be subject to the above processing in providing all of the resource deficiencies. Even though the task is set aside until the resources are released, the resources are not committed to or reserved for this task. It can well happen that the incoming task will remain in a backorder status because the released resources are used by other, higher-priority tasks coming along later.

Task Preemption

Task preemption may interrupt in-process tasks of lower priority to provide the needed resources to higher priority tasks. The number of in-process tasks that are interrupted or preempted depends upon the priority category for the incoming task. As examples, highest priority tasks can be allowed to preempt as many in-process tasks as necessary to recover the needed resources, while low priority tasks may not be permitted to preempt at all. The input data specify the maximum number of tasks each priority group can preempt.

Tasks with the lowest priorities are preempted first. Each preempted task is placed in a backorder status, and its remaining processing time is increased by a specified factor to represent the additional time involved in starting the task when resources again become available. However, tasks are preempted only if there are enough meeting the selection criteria to satisfy all resource deficiencies for the incoming task. If so, the preempted tasks are backordered and the incoming task is started; otherwise, the incoming task is backordered.

In both the expedited repair and preemption procedures, substitute items as well as prime items are considered. Thus, in-process tasks may be expedited or preempted for combinations of prime and substitute items, so long as sufficient amounts of both are obtained to meet the requirements. In no case, however, can both expediting and preempting be used.

TASK COMPLETION PROCEDURES

When a task is completed, temporary resources are released and may be reused for other tasks. At this time, a search is made to determine whether other tasks are in a backorder status for the resources being released. If there are some, they are ranked by priority and the task starting procedure is applied, commencing first with the highest priority task in the list. Before being ranked and selected for this processing, however, a priority escalation procedure is applied to the backordered tasks.

The escalation procedure increases the priority of tasks that have been in a backorder status for quite a while so that they can compete better for the resources being released. Otherwise, low priority tasks could remain in a backorder status throughout the simulation. The time a task must be backordered before its priority is increased is specified by an input factor according to priority category.

After priorities have been adjusted by the escalation procedure, the task starting procedure is applied to the backordered tasks just as if they were new tasks being generated. This is done because the situation has changed with respect to the use of substitutes, expedited repair, and preemption since the tasks were first backordered. Thus, even though released resources may be applied against backordered tasks, further opportunities for starting backordered tasks may now be available. The starting procedure uses both the released resources and the other possibilities in satisfying requirements of backordered tasks.

If the task being completed generates a resource, the model establishes a record to represent the resource. Reference is made to the Control Table to determine what task or tasks the resource incurs next as it commences its own path through the network as an independent entity. For example, the task being completed may represent the removal of a component from an airplane; this component then undergoes a set of shop repair tasks as prescribed by the network. The maintenance personnel, AGE, or other resources temporarily used in the removal task are, of course, released for further use as described above.

After the above processing is accomplished for a task being completed, reference is made to the Control Table to determine which, if any, tasks should be processed next. The starting procedure, as previously described, is then applied to these new tasks. In this procedure, a newly generated task may preempt a previously backordered task that has just started. This happens rarely, however, because either the resource contents of the several tasks involved differ or the new task has the same or lower priority than the previously backordered task. On the other hand, the procedures that apply at the start and end of a task interrelate, and a variety of complex situations can result.

CANNIBALIZATION

When a mission requirement occurs and there are not enough airplanes available, a cannibalization procedure may be applied to acquire the needed airplane or airplanes. In this procedure, a search is made for airplanes that are inoperative because they lack parts. If an airplane is missing not more than two parts and is otherwise serviceable, the cannibalization procedure obtains the needed parts from a down airplane that is missing the most parts other than those needed. This procedure consolidates parts shortages into one airplane. However, an airplane with more than five parts shortages will not be considered as a candidate for cannibalization.

Appropriate tasks are then generated to represent the removal of the parts from the cannibalized airplane and their installation on the airplane being prepared for the mission. If these tasks are accomplished in time, the airplane is used for the mission. Otherwise, the mission may be canceled and the airplane used later. Meanwhile, the cannibalized airplane might remain in an unserviceable status a longer time because it now lacks more parts.

SHIFT CHANGES

The shift change procedure causes increases or decreases in resource availabilities at specified times during the simulation. This procedure can represent changes in manning levels for different shifts during the day, as well as changes in other resource levels. The Shift Tables contain data prescribing what resources are affected, when their levels are changed, and by how much. These tables may represent several shift policies, and give a list of resources subject to each policy. Each policy has its own shift pattern that prescribes when shift changes occur. A shift pattern consists of several shift intervals and instructions concerning their repetition. Each interval and resource subject to the shift policy is assigned an authorized level in the Shift Table data. Figure 12 illustrates these data describing the shift operation. In this example, there are five different shift intervals. The first three, each 8 hours long, represent the three normal weekday shifts. The last two, each 24 hours long, represent weekend shifts for all day Saturday and Sunday. The repetition instructions prescribe the application of the first three shift intervals five times in succession to represent the five weekdays. Following this, shift intervals 4 and 5 are enacted to represent the two weekend shifts. Upon completion of shift interval 5, the whole pattern is repeated for the following week, and so forth. For each shift interval, authorized levels are given for each affected resource. Resource C, for example, might represent a maintenance specialty with 3, 4, and 9 men authorized for the respective weekday shifts, 9 men for all day Saturday, and none for Sunday.

REPETITION INSTRUCTIONS:		DO $\boxed{5}$ TIMES;			THEN DO $\boxed{1}$ TIME	
SHIFT INTERVALS:		1	2	3	4	5
SHIFT DURATIONS:		8 hours	8 hours	8 hours	24 hours	24 hours
AUTHORIZED RESOURCE LEVELS:	A	10	8	6	5	0
	B	5	5	5	8	2
	C	3	4	9	9	0
	D	8	8	6	5	0
	E	4	2	0	6	0

Fig. 12 -- Example shift policy data

Whereas a shift pattern similar to Fig. 12 might apply to maintenence personnel, different patterns might be defined for other kinds of resources. For example, the shift pattern in Fig. 13 might apply to aircraft in the simulation. In this example, the authorized strength for aircraft type A is 10 during the first four weeks, and 18 thereafter. The authorized level for aircraft type B starts at 5, builds up to 20 in the fourth week, and remains at 20 thereafter. The authorized number of aircraft type C is 18 except that every third week after the first four weeks 9 are withdrawn for some other duty. This shift policy, although structurally the same as that illustrated in Fig. 11, indicates the kinds of complexities that shift procedures can encompass with respect to resource availability. A variety of such shift policies, each applicable to a prescribed set of resources, may operate simultaneously during the simulation.

Whenever a shift change occurs for a given resource, the resource level will be either increased, decreased, or left the same, as prescribed by the authorized levels in the shift tables. The shift

REPETITION INSTRUCTIONS:		DO $\boxed{1}$ TIME;			THEN DO $\boxed{99}$ TIMES	
SHIFT INTERVALS:		1	2	3	4	5
SHIFT DURATIONS:		2 weeks	1 week	1 week	2 weeks	1 week
AUTHORIZED LEVELS:	A	10	10	10	18	18
	B	5	10	20	20	20
	C	18	18	18	18	9

Fig. 13 -- Example shift policy data for airplanes

change procedure takes no action if the level remains the same. If it is increased, the on-hand balance for the resource is appropriately increased and the task completion procedure, as previously described, is applied; the additional amounts gained by the shift change are logically equivalent to the resources released by a completed task. The task completion procedure determines whether there are backordered tasks for the resource and, if so, as many as possible with the additional resource supply provided by the shift change.

If the resource level is decreased, a check is made to determine whether enough of the resource is on-hand (not being used by tasks) to provide the decrement. If so, the on-hand balance is correspondingly reduced and no further action is taken. However, if part or all of the reduction cannot be taken from on-hand assets, the required amount must be recovered from tasks in process that are using the resource.

The recovery procedure inspects the completion times of in-process tasks using a resource. Tasks that will finish within a specified time are permitted to continue, and the resources they use are considered as satisfying part or all of the amount to be recovered. The time allowed depends upon the priority group to which the task belongs; higher priority tasks may be permitted to continue longer than lower priority ones. This procedure is designed to represent an overtime policy, wherein resources such as maintenance personnel are permitted to temporarily exceed their authorized levels. When tasks subject to the overtime policy finish, appropriate amounts of the released resources are used to satisfy the reduction in levels corresponding to the new shift.

If the resource level reductions are not fully satisfied by the overtime policy, tasks are preempted (interrupted) in order to recover the needed amounts. Lowest priority tasks are preempted first and placed in a backorder status until the released resources satisfy the shift reduction requirements. These tasks will be restarted when resources become available either through completion of other tasks or by increases in levels caused by a later shift change.

In general, the shift change procedure is similar to the task start and completion procedures, in that it also needs resources when released. The overtime policy of the shift change procedure is analogous to the expedited repair policy of the task start procedure. The main difference is that substitutes are not involved in the shift change procedures, whereas they are important in the task start and completion procedures.

PERFORMANCE SUMMARY REPORT

The Performance Summary Report is the primary output of the simulation program. It contains 65 overall performance statistics divided into the following functional groups:

Operations
Aircraft
Personnel
Shop Repair
Supply
Equipment

Referring to the example in Fig. 14, there is one row of data for each of the 65 statistics. Aside from the "TOTAL" column, the columnar readings may be different for each of the six groups. In fact, the user can control (on Input Form 10--Performance Summary Report) what is printed as columnar headings and the meaning of the data under them, within certain limits.

In general, statistics are accumulated during the simulation and assigned to a particular column of the report according to parameter value. Generally, the parameter is a column number assigned to each resource item in the Resource Table. If, for example, the user wants the columns for the supply portion of the report to represent groupings by the first digit of a part identifier code, he assigns each supply item a column number under which the item's supply statistics are to be accumulated. If unit cost or demand rate categories are desired, then the appropriate category number may be assigned to each supply item, causing its statistics to be accumulated and presented in the corresponding column of the report.

Similar procedures apply to the other groups of statistics. Manpower statistics may be grouped by shop (as illustrated in the example), by skill type, by AFSC groupings, and so forth. Aircraft statistics may be distributed by squadron and by aircraft type, for example. Possibilities of these kinds are limited only to the extent that resource items can be assigned to columns of the report. There is no limit on the number of columns that may be used, except that imposed by the computer storage capacity. If more than ten columns are used, the remaining columns will appear on successive pages.

The main exception to this procedure is the Operations group statistics. Here, only mission type controls the columnar split-out. The user, however, may still define mission types as he pleases and label the column headings to correspond.

For any given run of the simulation program, the Performance Summary Report may be produced in two levels. The Level I reports are produced on a periodic basis as specified by the user: every day, every other day, every week, etc. Statistics in this report represent performance during the indicated period. The Level II reports are produced for a given multiple of the Level I reports and contain summary statistics over a longer period. Thus, Level I reports may be produced each day, with Level II reports being produced weekly; or Level I reports may appear weekly, with Level II reports containing monthly summaries. Level II and Level I formats are identical.

OTHER OUTPUT REPORTS

When requested by appropriate entries on Form 1 (Run Specifications), the simulation program can produce one or more of the following output reports:

Resource Status
Mission Status
In-Process Status
Backorder Status
Diagnostic

The first four reports contain data representing the status of the simulated environment at particular points in simulated time as specified by the user on Form 1. The last report (Diagnostic) is used by a programmer or experienced user of the model as an aid in locating suspected malfunctions of the model. It, in effect, lists data for each transaction that occurs between two points in simulation time, as specified on Form 1.

```
RUN NUMBER TANGO          P E R F O R M A N C E   S U M M A R Y                              PERIOD FROM 41.   TO 42.

        O P E R A T I O N S           TOTAL     HDEL    LDEL    NCEL

 1  NUMBER OF MISSIONS REQUESTED      30.        12.     15.      3.      0.
 2        NUMBER ACCOMPLISHED         27.        12.     12.      3.      0.
 3        PERCENT ACCOMPLISHED        90.00     100.     80.00  100.      0.
 4  NUMBER OF SORTIES REQUESTED       49.        24.     22.      3.      0.
 5        NUMBER ACCOMPLISHED         45.        24.     18.      3.      0.
 6        PERCENT ACCOMPLISHED        91.84     100.     81.82  100.      0.

        A I R C R A F T               TOTAL     COCOA1  COCOA2  TANGO1  TANGO2

 7  NUMBER OF AIRCRAFT AUTH. (EOP)    36.        0.      0.     18.     18.      0.
 8  NUMBER OF AIRCRAFT-DAYS AVAIL.    36.        0.      0.     18.     18.      0.
 9        PCT SORTIES (INCL ALERT)     9.63      0.      0.      9.51    9.76    0.
10        PCT UNSCHED MAINTENANCE     39.98      0.      0.     38.73   41.24    0.
11        PCT SCHED MAINTENANCE       10.26      0.      0.      9.37   11.16    0.
12        PCT NORS                     8.45      0.      0.      8.56    8.33    0.
13        PCT SERVICE + MSN. WAIT      4.06      0.      0.      4.26    3.86    0.
14        PCT OPERATIONALLY READY     27.61      0.      0.     29.57   25.66    0.
15  AVG. AIRCRAFT TURNAROUND TIME      9.12      0.      0.      8.16   10.08    0.
16  AVG. NO. OF SORTIES/ A/C /DAY      1.25      0.      0.      1.22    1.28    0.

        P E R S O N N E L             TOTAL     301XX   422XX   431X1   431X2   431XX   462XX   432X0   322X1   461X0

17  MANHOURS AUTHORIZED (100)         54.00      2.72    2.56    9.68   12.16    2.08    7.28    8.00    2.72    0.48    6.32
18  MANHOURS AVAILABLE (100)          54.00      2.72    2.56    9.68   12.16    2.08    7.28    8.00    2.72    0.48    6.32
19  PERCENT UTILIZATION               30.14     38.76   67.98   34.20    3.30   23.92   24.66   33.47   54.76    3.95   52.04
20  MANHOURS USED (100)               16.27      1.05    1.74    3.31    0.40    0.50    1.79    2.68    1.49    0.02    3.29
21        PCT UNSCHED. MAINTENANCE    78.66    100.     90.56   32.48    0.    100.     99.06   76.34  100.    100.     99.32
22        PCT SCHED. MAINTENANCE      21.34      0.      9.44   67.52  100.      0.      0.94   23.66    0.      0.      0.68
23  NUMBER OF MEW DEMANDED          1193.       69.    105.    324.     24.     43.    264.     52.    120.      2.    190.
24        PCT AVAILABLE (PRIME)       63.12     46.38   34.29   74.38  100.     65.12   87.88  100.     40.00  100.     30.53
25        PCT AVAILABLE (SUBST.)       0.        0.      0.      0.      0.      0.      0.      0.      0.      0.      0.
26        PCT PROV. BY EXPEDITE        2.18      5.80    4.76    0.      0.      0.      5.30    0.      1.67    0.      0.53
27        PCT PROV. BY PREEMPTION      1.42     13.04    1.90    0.      0.      0.      0.      0.      2.50    0.      1.58
28        PCT DEMANDS NOT SATIS.      33.28     34.78   59.05   25.62    0.     34.88    6.82    0.     55.83    0.     67.37
29  OVERTIME MANHOURS USED (100)       0.03      0.      0.01    0.      0.      0.      0.      0.      0.01    0.      0.00
30  MANHOURS PER FLYING HOUR          19.55      1.27    2.09    3.98    0.48    0.60    2.16    3.22    1.79    0.02    3.95
31  MOST TROUBLESOME PERS. ITEMS       0.      486.01  485.03  485.02  485.01  484.03  484.02  484.01  483.03  483.02  483.01
```

Fig. 14 -- Example of performance summary report

```
RUN NUMBER TANGO                P E R F O R M A N C E    S U M M A R Y                        PERIOD FROM  41.   TO  42.

           S H O P   R E P A I R         TOTAL     11-14   ENGINE   41-44    45-47    51-52    71-73   SYS 74   SYS 75   76-77

32  NO. OF REPARABLE GENERATIONS          68.        20.      0.       2.       3.       2.      18.       9.      11.      2.        1.
33        PCT BASE REPAIR                 75.      65.00      0.      50.     66.67     50.     83.33    66.67    90.91    100.      100.
34        PCT DEPOT REPAIR                25.      35.00      0.      50.     33.33     50.     16.67    33.33     9.09     0.        0.
35  AVERAGE BASE REPAIR CYCLE              0.22     0.13     0.64     0.36     0.71     0.       0.21     0.20     0.17    0.18      0.
36        PCT ACTIVE REPAIR              100.      100.     100.     100.     100.       0.     100.     100.     100.     100.        0.
37        PCT WHITE SPACE                  0.        0.       0.       0.       0.       0.       0.       0.       0.      0.        0.
38  NO. OF ITEMS IN REPAIR (EOP)           8.        0.       0.       0.       1.       1.       2.       2.       2.      0.        0.
39  NO. OF ITEMS BACKLOGGED (EOP)        318.      124.       2.      17.      31.      29.      58.      42.      14.      0.        1.

           S U P P L Y                   TOTAL     11-14   ENGINE   41-44    45-47    51-52    71-73   SYS 74   SYS 75   76-77

40  TOT DOLLAR INVEST.(1000)(EOP)      6576.19    619.24 3057.60   160.16   123.73   317.24 1289.51   679.99   262.29    38.70     27.72
41  FILL RATE PERCENT                    91.18    100.       0.     100.     100.     100.      77.78    88.89    90.91   100.      100.
42  NUMBER OF BACKORDER-DAYS              4.50      0.       0.       0.       0.       0.       1.76     1.42     1.31    0.        0.
43  NUMBER OF UNITS DEMANDED             68.       20.       0.       2.       3.       2.      18.       9.      11.      2.        1.
44        PCT OFF-THE-SHELF              91.18    100.       0.     100.     100.     100.      77.78    88.89    90.91   100.      100.
45        PCT EXPEDITED REPAIR            0.        0.       0.       0.       0.       0.       0.       0.       0.      0.        0.
46        PCT PREEMPTION                  0.        0.       0.       0.       0.       0.       0.       0.       0.      0.        0.
47        PCT DEMANDS NOT SATIS.          8.82      0.       0.       0.       0.       0.      22.22    11.11     9.09    0.        0.
48  NUMBER OF CANNIBALIZATIONS            1.        0.       0.       0.       0.       0.       1.       0.       0.      0.        0.
49  NO. ITEMS ON BACKORDER (EOP)          3.        0.       0.       0.       0.       0.       1.       1.       1.      0.        0.
50                                        0.        0.       0.       0.       0.       0.       0.       0.       0.      0.        0.
51  MOST TROUBLESOME SUPPLY ITEMS         0.       14.      13.      12.      11.      10.       9.       8.       7.      6.        5.

           E Q U I P M E N T             TOTAL    SYS 74   SYS 23   41-44    45-47    51-52    71-73    11-14   SYS 75   76-77

52  TOT DOLLAR INVEST.(1000)(EOP)      5745.91   1506.06     8.87    94.63    27.60   689.39 1963.15    84.58   284.40    77.87  1009.36
53  EQUIPMENT HOURS AUTH. (100)         186.96     11.04     0.72     4.80     0.72     6.      14.64    19.20     8.64     5.04   116.16
54  EQUIPMENT HOURS AVAIL. (100)        186.96     11.04     0.72     4.80     0.72     6.      14.64    19.20     8.64     5.04   116.16
55        PCT USED - UNSCHED MAINT        7.28      8.72     2.52     4.85     0.       3.77     9.19     4.50     5.22     1.54     8.13
56        PCT USED - SCHED MAINT          0.01      0.       0.       0.       0.       0.       0.       0.07     0.       0.       0.
57        PCT UNUSED                     92.71     91.28    97.48    95.15   100.      96.23    90.81    95.43    94.78    98.46    91.87
58  NUMBER OF BACKORDER-DAYS             10.66      0.00     0.       0.       3.       0.       1.47     3.19     0.       0.       5.99
59  NUMBER OF UNITS DEMANDED            633.       54.       1.       7.       1.      16.      63.      39.      26.       4.     422.
60        PCT AVAILABLE                  95.42    100.     100.     100.     100.     100.     100.      97.44   100.     100.      93.36
61        PCT PROV. BY EXPEDITE           0.        0.       0.       0.       0.       0.       0.       0.       0.       0.       0.
62        PCT PROV. BY PREEMPTION         0.63      0.       0.       0.       0.       0.       0.       0.       0.       0.       0.95
63        PCT DEMANDS NOT SATIS.          3.95      0.       0.       0.       0.       0.       0.       2.56     0.       0.       5.69
64  EQUIP. HOURS BACKLOG(100)(EOP)        1.61      0.       0.       0.       0.       0.       0.07     0.80     0.       0.       0.75
65  MOST TROUBLESOME EQUIP. ITEMS         0.      364.     363.     362.     361.     360.     359.     358.     357.     356.     355.
```

Fig. 14 -- Example of performance summary report (cont.)

V. EMBEDDED DECISION MODEL

In general, a simulation model can only be used to evaluate a particular study. The input data describe the actual or hypothetical environments and operating conditions; the output data provide the consequences of the assumed operation in terms of performance, resource utilization, and other measures. But in any particular run, the simulation model cannot arrive at decisions or produce best or preferred solutions to a particular problem. Only through a carefully designed sequence of runs, involving controlled changes in particular policy variables, can preferred solutions be approximated.

To help alleviate this difficulty and to reduce the number of runs required for certain kinds of study objectives, a special procedure has been incorporated in the Logistics Composite Model. This section describes the procedure, a mathematical decision model of separate subroutines embedded in the main simulation program.

PURPOSE

The main purpose of the embedded decision model is to help determine a best mix of resources to support a prescribed flying program within given performance tolerances. More specifically, the model changes resource levels whenever a designated performance measure falls to an unacceptable level during a simulation run. Levels are altered according to a cost-effectiveness criteria, where resources that increase system performance the most on a per dollar basis are considered first for increases in stockage.

The procedure might also be used to impose an overall dollar constraint on support resources. In this application, the decision model would be used to determine the mix of resources, within the dollar constraint, that maximizes system performance for a given flying program. The decision model reduces the number of simulation runs that would otherwise be required to solve this problem.

The decision process consists of two main parts, a forecasting procedure and the decision model itself. These features are optional for any given run; if used, however, corresponding forms must be completed to provide input data needed for their operation. The decision procedures and data requirements are further described below.

FORECASTING PROCEDURE

During simulation, a variety of system performance statistics are accumulated and output as a periodic performance report. Some of these statistics have been specifically designated for possible use in controlling the application of the embedded decision model. In particular, the following performance measures have been selected for this purpose:

Sortie effectiveness (sorties accomplished as a percentage of sorties requested)

Average aircraft turnaround time

Average number of sorties per aircraft per day

These measures are produced periodically during the simulation, with the period being fixed at one day for most runs. If daily observations for a particular statistic such as sortie effectiveness are plotted on a graph, the results may be similar to those shown in Fig. 15.

Data like those in Fig. 15 are called "Time Series," and can vary considerably from day to day. Since the decision model increases the levels of critical resources whenever the designated performance measure falls to a specified point, it is undesirable to react to the daily fluctuations. To avoid this, the fluctuations are "smoothed" by a forecasting procedure.

Each period, the forecasting procedure estimates the performance measure for the next period, using data observed (produced by the simulation model) so far. These forecasted values smooth the observed data; when plotted on a graph along with the observed data, they may appear as shown in Fig. 16.

This figure also shows a trigger level which controls the decision model. As the simulation progresses, the decision model is called into operation whenever the *forecasted* value of the designated

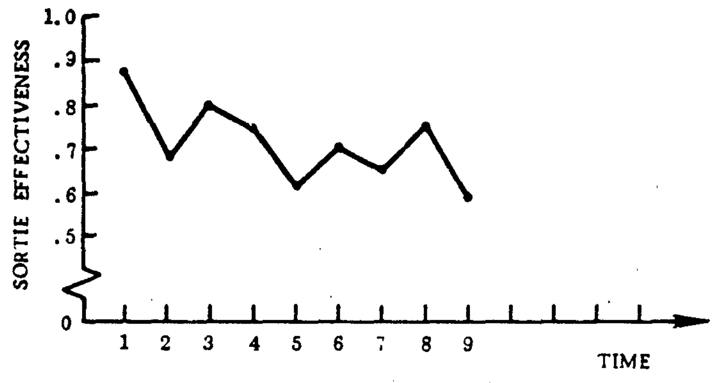


Fig. 15 -- Example of a time series

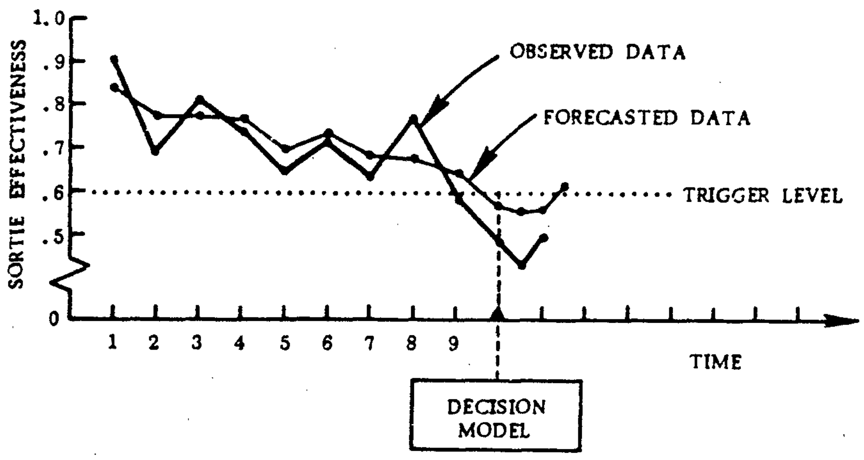
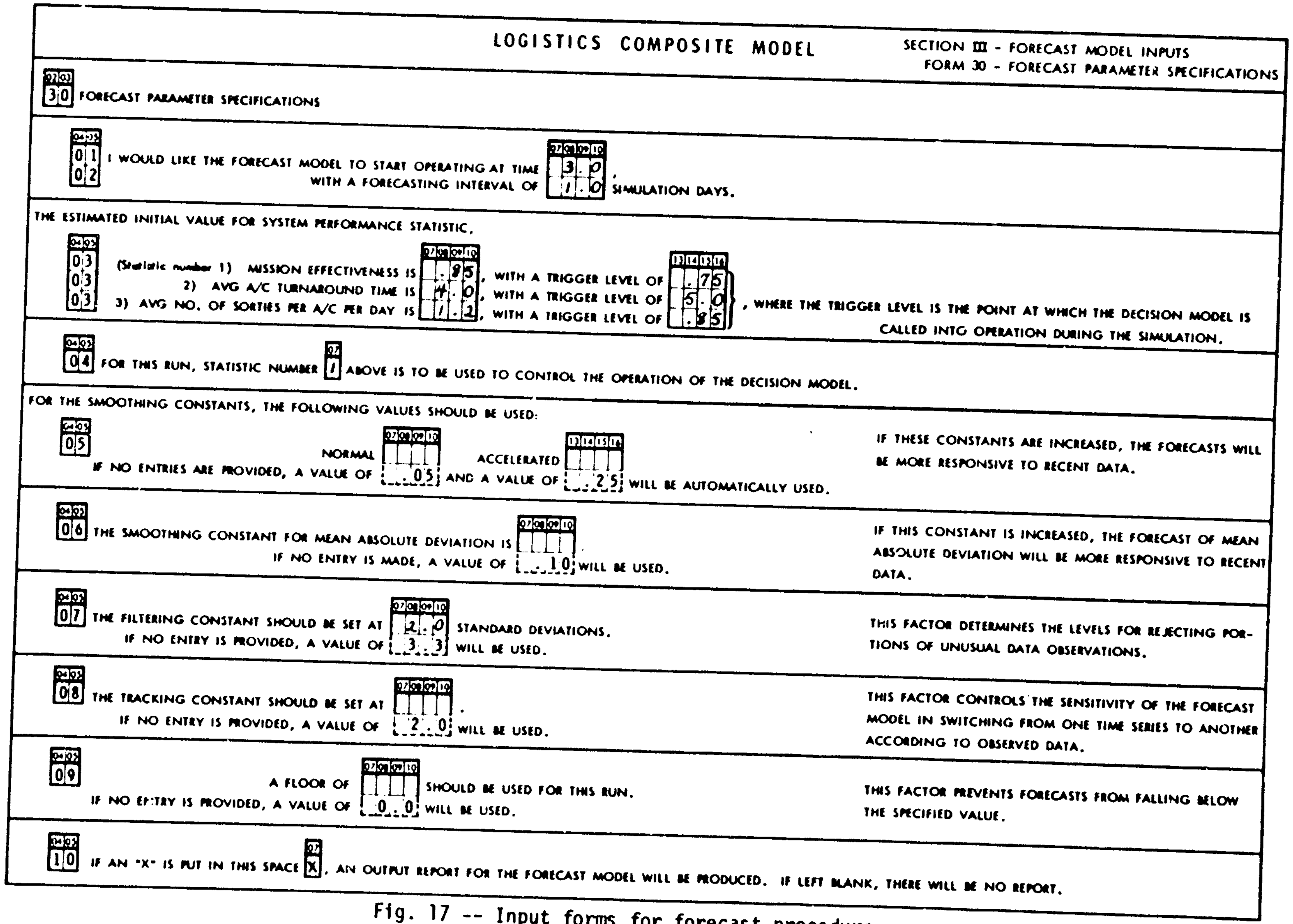


Fig. 16 -- Example of the forecasting procedure

performance measure falls to the trigger level. The user puts in the desired trigger levels for each performance measure listed above; for any given run, however, the only operative trigger level will be that for the particular performance measure controlling the decision process.

Input Form 30 (Forecast Parameter Specifications) is used to identify the performance measure controlling the decision process and to provide values for the trigger levels and other factors used in the forecasting procedure. This form, with example entries, is illustrated in Fig. 17.



LOGISTICS COMPOSITE MODEL — SECTION III - FORECAST MODEL INPUTS, FORM 30 - FORECAST PARAMETER SPECIFICATIONS

30 FORECAST PARAMETER SPECIFICATIONS

01 I WOULD LIKE THE FORECAST MODEL TO START OPERATING AT TIME [3.0],
02 WITH A FORECASTING INTERVAL OF [1.0] SIMULATION DAYS.

THE ESTIMATED INITIAL VALUE FOR SYSTEM PERFORMANCE STATISTIC,

03 (Statistic number 1) MISSION EFFECTIVENESS IS [.85], WITH A TRIGGER LEVEL OF [.75]
03 2) AVG A/C TURNAROUND TIME IS [4.0], WITH A TRIGGER LEVEL OF [5.0]
03 3) AVG NO. OF SORTIES PER A/C PER DAY IS [1.2], WITH A TRIGGER LEVEL OF [.85], WHERE THE TRIGGER LEVEL IS THE POINT AT WHICH THE DECISION MODEL IS CALLED INTO OPERATION DURING THE SIMULATION.

04 FOR THIS RUN, STATISTIC NUMBER [1] ABOVE IS TO BE USED TO CONTROL THE OPERATION OF THE DECISION MODEL.

FOR THE SMOOTHING CONSTANTS, THE FOLLOWING VALUES SHOULD BE USED:

05 NORMAL [ ] ACCELERATED [ ]
IF NO ENTRIES ARE PROVIDED, A VALUE OF [.05] AND A VALUE OF [.25] WILL BE AUTOMATICALLY USED.
IF THESE CONSTANTS ARE INCREASED, THE FORECASTS WILL BE MORE RESPONSIVE TO RECENT DATA.

06 THE SMOOTHING CONSTANT FOR MEAN ABSOLUTE DEVIATION IS [ ].
IF NO ENTRY IS MADE, A VALUE OF [.10] WILL BE USED.
IF THIS CONSTANT IS INCREASED, THE FORECAST OF MEAN ABSOLUTE DEVIATION WILL BE MORE RESPONSIVE TO RECENT DATA.

07 THE FILTERING CONSTANT SHOULD BE SET AT [2.0] STANDARD DEVIATIONS.
IF NO ENTRY IS PROVIDED, A VALUE OF [3.3] WILL BE USED.
THIS FACTOR DETERMINES THE LEVELS FOR REJECTING PORTIONS OF UNUSUAL DATA OBSERVATIONS.

08 THE TRACKING CONSTANT SHOULD BE SET AT [ ].
IF NO ENTRY IS PROVIDED, A VALUE OF [2.0] WILL BE USED.
THIS FACTOR CONTROLS THE SENSITIVITY OF THE FORECAST MODEL IN SWITCHING FROM ONE TIME SERIES TO ANOTHER ACCORDING TO OBSERVED DATA.

09 A FLOOR OF [ ] SHOULD BE USED FOR THIS RUN.
IF NO ENTRY IS PROVIDED, A VALUE OF [0.0] WILL BE USED.
THIS FACTOR PREVENTS FORECASTS FROM FALLING BELOW THE SPECIFIED VALUE.

10 IF AN "X" IS PUT IN THIS SPACE [X], AN OUTPUT REPORT FOR THE FORECAST MODEL WILL BE PRODUCED. IF LEFT BLANK, THERE WILL BE NO REPORT.

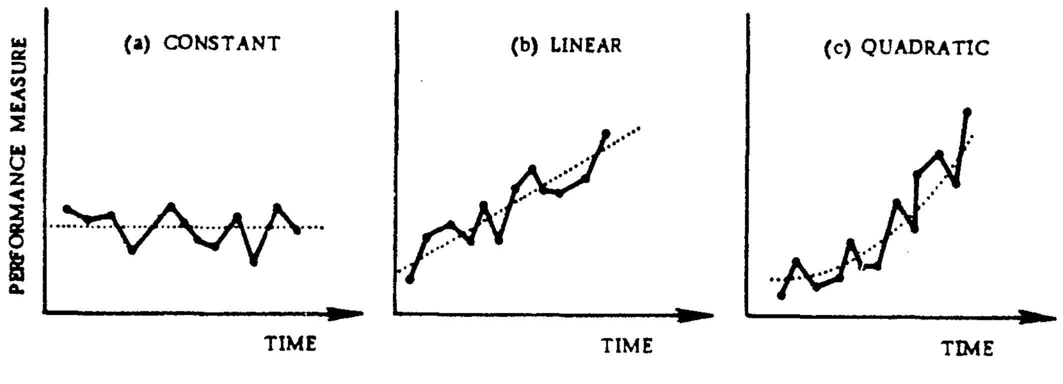
Fig. 17 -- Input forms for forecast procedure

In summary, the forecasting procedure smooths out the short term fluctuations in designated performance data in order to prevent the decision model from raising resource levels too often. More specific features of the forecasting procedure are described below.

Time Series Models

The forecasting procedure is based upon an exponential smoothing technique, in which the observed data are assumed to represent random fluctuations caused by an underlying process called a time series model. The kinds of time series models used by the forecasting procedure are a constant model, a linear model, and a quadratic model, illustrated in Figs. 18a, 18b, and 18c, respectively.

In the constant time series model (Fig. 18a), it is assumed that no real change in the observed data is expected; only random fluctuations about some constant value. In the linear model, a trend is expected--it may be upward, as shown in Fig. 18b, or downward. In either case, the observed data over time are assumed to represent random fluctuations about the upward or downward sloping line. In the quadratic model, an upward (or downward) trend is expected, but at an increasing (or decreasing) rate. Again, the observed data would be random variations about a curve like that shown in Fig. 18c.



Figs. 18a, b, and c -- Time series models

Only one time series model applies to a statistic processed by the forecasting procedure for any given period. However, the procedure may automatically switch from one model to another during simulation, depending upon the nature of the observed data. For example, a statistic such as mission effectiveness may be subject to the constant model to start with. Then, as the simulation progresses, a downward trend may occur. The forecasting procedure detects this trend from the observed data and if it appears significant, automatically causes a change from the constant model to a linear model. Using the linear model, then, a better forecast will result since the trend will now be explicitly considered. Similarly, a switch to a quadratic model may be made if a significant change in trend is detected. If later in the simulation, a trend or change in trend is no longer present, the forecasting procedure will switch back to a lower order time series model. This switching feature applies to each system performance measure independently.

Filtering

Unusual observations of the statistic being forecast are partially rejected for consideration in the forecasting procedure. This is illustrated in Fig. 19, where the shaded areas represent the observed data that are rejected. In this example, the underlying time series model is assumed constant. The dashed lines represent upper and lower limits for accepting the observations. The user may change these limits to reject more or less of the observed data as he desires.

The filtering procedure prevents the forecasting procedure from over-reacting to occasional or sporadic data observations of an unusual nature. These unusual situations may be caused by a particular combination of random events occurring during the simulation and may be unrepresentative of the general processes being simulated.

Accelerated Smoothing

During simulation, the observed data may show a sudden upward or downward jump or step. For example, mission requirements may suddenly be doubled, causing a sharp drop in mission effectiveness. Assuming the performance measure is governed by a constant time series model, this situation is shown in Fig. 20.

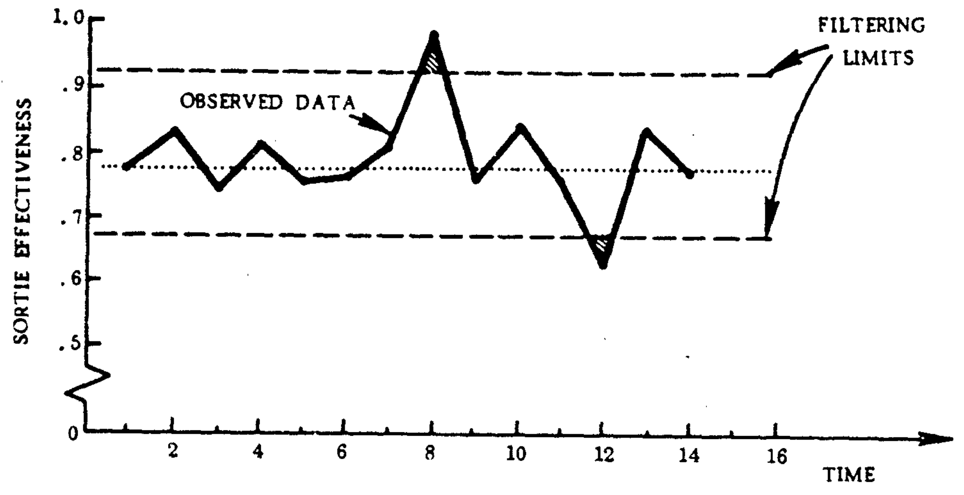


Fig. 19 -- Example of the filtering process

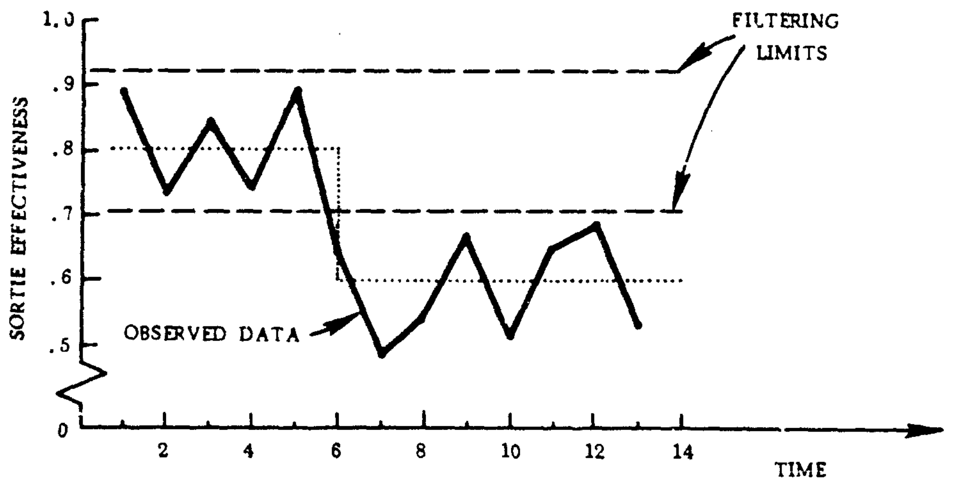


Fig. 20 -- Example of accelerated smoothing

In this example, the sudden drop occurs in period 6, where the observed da˙a drop below the previously applicable filtering limits. The forecasting procedure detects this kind of change by keeping track of the number of times the filtering limit has been exceeded. When the observed data exceed the filtering limit twice in a row, it is assumed that this does not occur by chance (since the probability of it happening by chance is quite small), but indicates that a true alteration of the underlying process has occurred.

When a change of this kind is detected, the forecasting procedure automatically switches to "accelerated" smoothing. In effect, a smoothing constant, used in the forecasting procedure to control the forecast's responsiveness to observed data, is greatly increased so that more recent data will have a much greater influence upon the forecast than older data. In this way, the forecast "homes in" on the new situation faster than it otherwise would. When the accelerated smoothing is no longer needed, the smoothing constant changes back to its former value. This feature applies when the statistic is governed by a linear or quadratic time series model as well as by the constant model illustrated in Fig. 20.

Time of Start

For any given simulation in which the forecasting procedure is applied, the user may specify the simulation time at which the model first becomes operative. This may be time zero or any time thereafter.

This feature enables the user to skip over the first few periods of the simulation where the results are usually not valid because of the way the data are initialized. For example, the simulation starts with all resources available up to their authorized allowances. Until jobs occur to use the resources, the performance statistics will be unrealistic and should not be used in the forecasting procedure. The simulation will usually "settle down" after a few simulated days, at which time the forecasting procedure may be applied.

Floor

The forecasting procedure has a number called the "floor" which is set by the user to always keep the forecasted statistics above that value. Whenever a data observation falls below the floor, the floor itself is substituted for the observation, thereby always producing a forecast above this value.

DECISION MODEL

During simulation, the forecasting procedure may determin that a designated performance measure has reached a critical or "trigger" level, representing a lower bound for acceptable performance. At this time, the decision model is called into operation in order to relieve the situation and to bring the performance measure back to acceptable values. The decision model accomplishes this by raising authorized levels of the resource items causing the most trouble.

For the decision model to determine what resource items are most likely causing the drop in system performance, a measure is defined for each item indicating how well it is meeting requirements. The derivation of this measure, or "utilization index," may be explained by referring to Fig. 21, in which the balance of a particular item is plotted as a function of simulated time. Positive values indicate

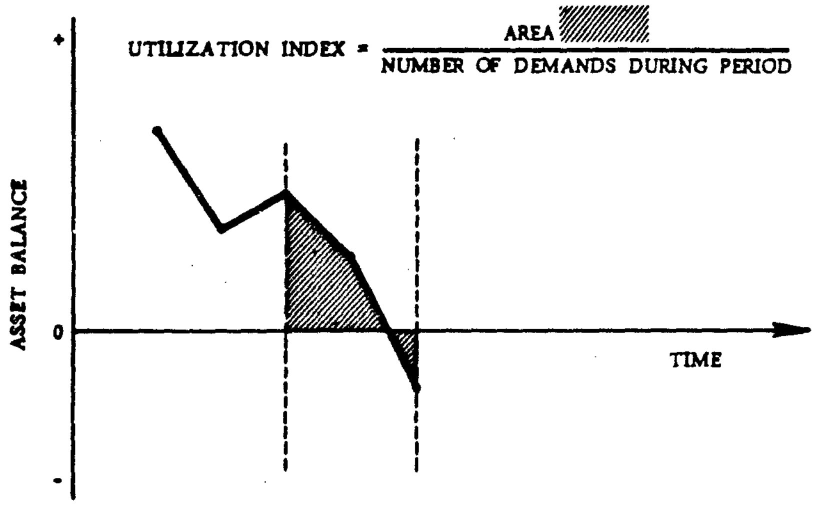


Fig. 21 -- Utilization index

that supply of the item is larger than demand. Negative values indicate due-outs--instances where demand has exceeded the available supply.

If a particular interval is defined, as indicated by the vertical dashed lines in Fig. 21, then the area under the balance graph, indicated by shading in Fig. 21, may be computed. If this area is positive, then the item is in a relatively good position with respect to meeting requirements. If the area is negative, it indicates that the item is failing to meet requirements. This area, divided by the number of demands during the period, represents the item's utilization index.

For each resource item, then, the utilization index is computed each period. This index may vary considerably from one period to another, perhaps fluctuating between negative and positive over simulated time. When the decision model is called into operation, it may happen that an item is in serious trouble (large negative value of the utilization index) at that time by sheer chance and that, normally, the supply of the item can meet requirements. To prevent the decision model from reacting to these short-term fluctuations, the item utilization indices are therefore smoothed by a simplified version of the technique used by the forecasting procedure. This version performs single exponential smoothing only, where the underlying time series model is assumed constant.

The decision model, then, uses the smoothed values of the utilization indices in determining which items are "in trouble" and presumably degrading overall system performance. The indices themselves, however, are only one kind of factor the decision model uses in determining which resource items to augment and by how much. Another factor is the cost of adding one unit of the resource to the available inventory. If two resource items have the same utilization index, but one is cheaper than the other, the decision model will add a unit of the cheaper item to the inventory if a choice must be made, since the same increase in system performance may be expected at less cost.

In general, the decision model determines which items should have their levels increased according to a "utility measure," computed as an exponential function of the utilization index for each item divided by its unit cost. In addition, the decision model allows for the fact that the second, third, fourth, etc., units added to the inventory of a given item will be of successively less value with respect to increasing overall system performance. This factor is considered by multiplying the utility measure for the first added unit by a fraction to obtain a utility measure for the second added unit; this in turn is multiplied by the same fraction to obtain a utility measure for the third added unit, and so forth, as illustrated in Fig. 22. In this figure, the fraction used to obtain the utility

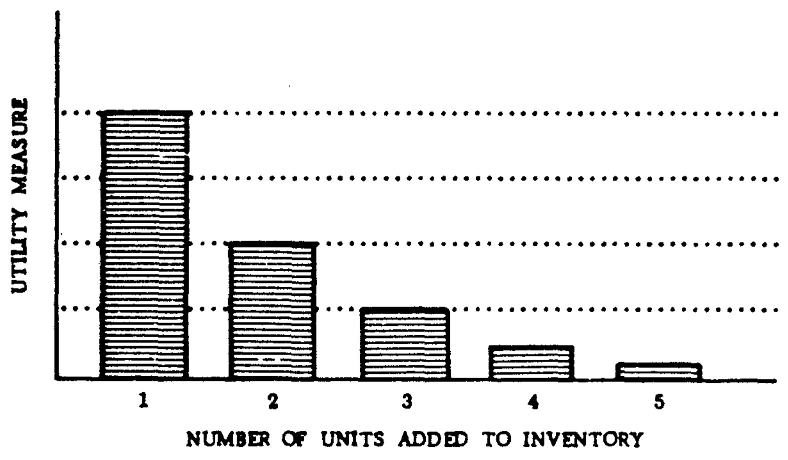


Fig. 22 -- Utility measures for additional units of an item

measures is one-half. The user may adjust the fraction, however, to obtain different relative values for successively larger amounts added to the inventory of a resource item.

Having computed the utility measure for each resource item and for each unit that might be added to inventory, the decision model then ranks all the measures from highest to lowest. The results might be as shown below. With this example, the decision model would first increase the level of resource 5 by 1 unit, then resource 8 by 1 unit, then resource 2 by 1 unit, then resource 5 by a second unit, and so

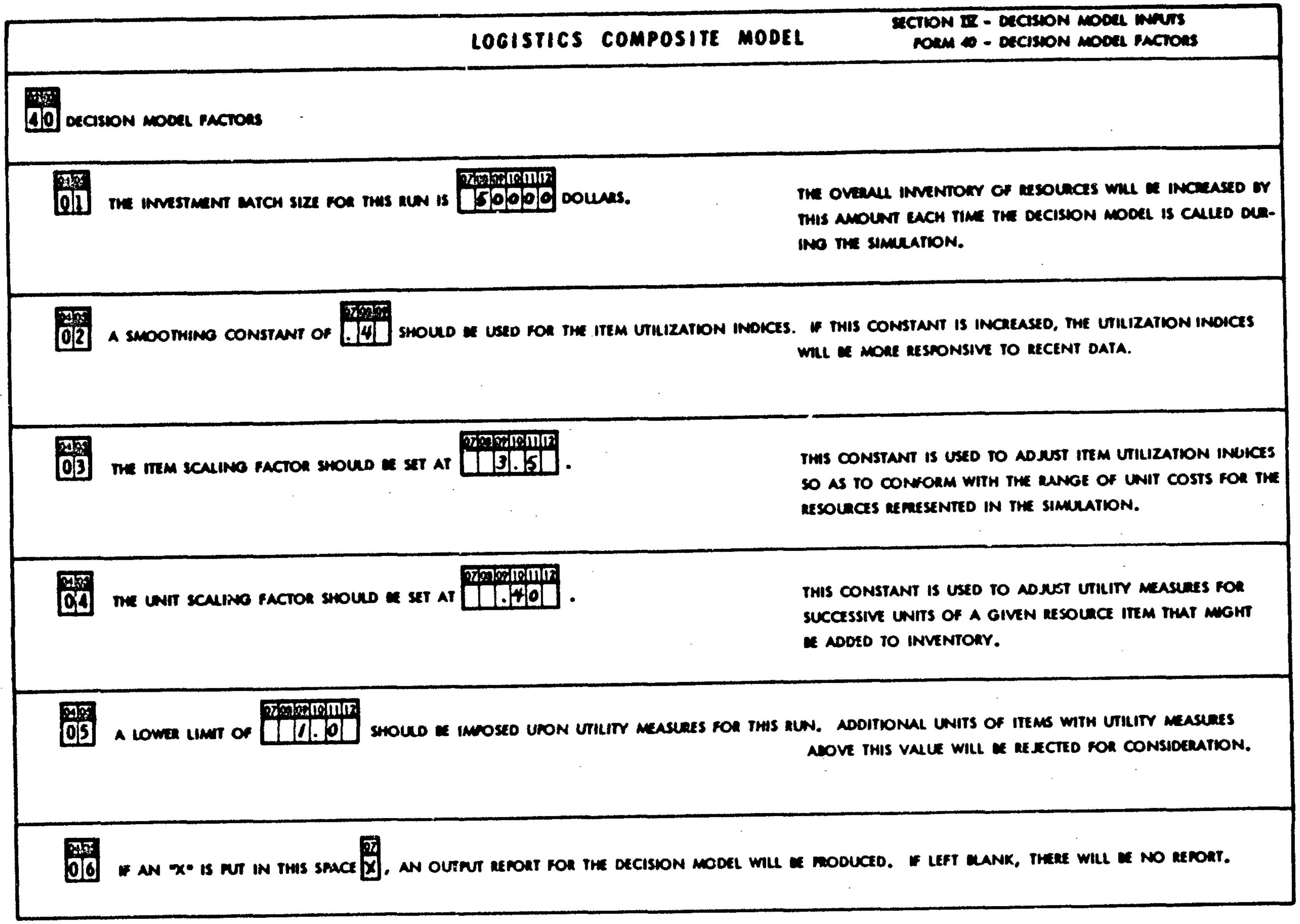
Resource No.	Unit	Utility Measure
5	1	2.3
8	1	2.2
2	1	1.9
5	2	1.8
3	1	1.5
1	1	.9
8	2	.3
7	1	.1
5	3	.05
6	1	.02

on, working down the list in succession. Since the utility measure indicates the relative "worth" of an additional unit of resource, with respect to increasing system performance, units with highest values are assigned first, followed by units with successively lower values. Input Form 40 (Decision Model Factors) provides factors that the decision model uses. This form, with example entries, is illustrated in Fig. 23.

In summary, the decision model determines which resource items are in the most trouble per dollar of additional inventory investment and increases their levels. The simulation model then operates with the increased levels with the expectation that the degraded system performance will now be alleviated. More specific features of the decision model are described below.

Shift Changes

If a resource such as manpower is subject to shift changes, whereby authorized levels are changed over time, the decision model treats each different shift for each such resource as a separate resource item. By so doing, the model can detect which shifts are causing problems with respect to resource availability and cause levels to be raised on a shift basis. For example, the night shift for a particular shop may have inadequate manning, in which case the decision model may increase the authorized level, leaving the day shifts unchanged if they are adequately manned.



LOGISTICS COMPOSITE MODEL — SECTION IX - DECISION MODEL INPUTS, FORM 40 - DECISION MODEL FACTORS

40 DECISION MODEL FACTORS

01 THE INVESTMENT BATCH SIZE FOR THIS RUN IS 50000 DOLLARS. — THE OVERALL INVENTORY OF RESOURCES WILL BE INCREASED BY THIS AMOUNT EACH TIME THE DECISION MODEL IS CALLED DURING THE SIMULATION.

02 A SMOOTHING CONSTANT OF .4 SHOULD BE USED FOR THE ITEM UTILIZATION INDICES. IF THIS CONSTANT IS INCREASED, THE UTILIZATION INDICES WILL BE MORE RESPONSIVE TO RECENT DATA.

03 THE ITEM SCALING FACTOR SHOULD BE SET AT 3.5 . — THIS CONSTANT IS USED TO ADJUST ITEM UTILIZATION INDICES SO AS TO CONFORM WITH THE RANGE OF UNIT COSTS FOR THE RESOURCES REPRESENTED IN THE SIMULATION.

04 THE UNIT SCALING FACTOR SHOULD BE SET AT .40 . — THIS CONSTANT IS USED TO ADJUST UTILITY MEASURES FOR SUCCESSIVE UNITS OF A GIVEN RESOURCE ITEM THAT MIGHT BE ADDED TO INVENTORY.

05 A LOWER LIMIT OF 1.0 SHOULD BE IMPOSED UPON UTILITY MEASURES FOR THIS RUN. ADDITIONAL UNITS OF ITEMS WITH UTILITY MEASURES ABOVE THIS VALUE WILL BE REJECTED FOR CONSIDERATION.

06 IF AN "X" IS PUT IN THIS SPACE X, AN OUTPUT REPORT FOR THE DECISION MODEL WILL BE PRODUCED. IF LEFT BLANK, THERE WILL BE NO REPORT.

Fig. 23 -- Example of Input Form 40

Investment Batch

Each time the decision model is called into operation, the levels of some resource items are raised. The user may control the aggregate amount of increase, in terms of total dollars, by specifying an investment batch size on the appropriate input form. If, for example, an investment batch of $10,000 is specified, the decision model will add items to inventory up to a total of $10,000 worth and no more. In general, the smaller the batch size is, the more often one may expect the decision model to be called into operation. If the batch size is set too high, however, excess inventories may result.

Item Scaling Constant

When values of the utilization index are computed across all items, they will range from some negative value to some positive value, e.g., from -2 to +2. In deriving the utility measure, however, the utilization index is divided by the item's unit cost. The unit costs across items may range from, say, $100 to $100,000. For the allocation procedure to work properly, the utilization indices must be converted so that they will be spread out over about the same range of values as the unit costs. This is accomplished by using an exponential function that contains a scaling factor for multiplying the utilization index. Since the value for this "item scaling constant" depends upon typical ranges for the utilization indices and unit costs, several operational runs of the Logistics Composite Model must be made before it can be determined. Once it has been determined, however, it will probably not be necessary to alter it in subsequent uses.

Unit Scaling Constant

As described above, a constant fraction is used to calculate values of the utility measure for successive units of an item to be added to inventory. This fraction is controlled by a "unit scaling constant" that the user may adjust. As in the case of the item scaling factor, however, analysts may determine an appropriate value for this constant from early operational runs and then use the value for subsequent uses of the model.

Lower Limit for Utility Measure

A feature of the decision model is that items that meet requirements fairly well may not be considered in the procedure for raising levels. The user may input a number to represent a lower limit on the values of the utility measure that are used.

Referring to the tabulation on page 65, for example, the lower limit may be set to 0.1, in which case the last two items on the list (the third unit of resource No. 5 and the first unit of resource No. 6) will not be considered for inventory augmentation; in general, any unit of a resource with a utility measure less than 0.1 will be rejected in this case. As in the case of the scaling factors, this lower limit may require some initial experimentation to find an appropriate value.

OUTPUT REPORTS

At the option of the user, output reports showing results of the forecasting procedure and decision model may be produced each time these routines are exercised. Examples of these reports are shown in Fig. 24.

The forecasting report is divided into two sections. Section I contains the observed data during the previous forecasting period, the forecasts from the previous computations, and the new forecasts. Section II contains values for various parameters used in the forecast computations; these will interest only the analyst investigating the behavior of the forecasting procedure.

The output report for decision model results lists units added to inventory ranked according to their utility measures. For each unit added, various data elements are printed out as illustrated in Fig. 24. At the top of the report, assigned values for decision model parameters are shown for reference purposes.

```
RUN NUMBER  99999                F O R E C A S T    R E S U L T S            FORECAST PERIOD FROM   3.00 TO   4.00

     S E C T I O N   1  -  F O R E C A S T S

                                              FORECASTS               OBSERVED IN
     PERFORMANCE MEASURE               LAST PERIOD   THIS PERIOD       THIS PERIOD

MISSION EFFECTIVENESS                      .87           .85               .80

AVE. AIRCRAFT TURNAROUND TIME             4.67          4.93              5.25

AVE. NO. OF SORTIES PER A/C PER DAY       1.22          1.28              1.40


     S E C T I O N   2  -  P A R A M E T E R   V A L U E S

NO.   CVALU   FVALU   KAST LFLTR MULTA   SMAD    BASE    TREND   CHANG    SAVER   DAVER   TAVER

 1    .8046   .8511     1    1     1    0.1147 0.8476   .0213  0.0000   .8413   .8769   .9103

 2   5.2500  4.9304     1    2     2    0.5877 4.8918 -0.1672  0.0005  5.0076  5.3114  5.4192

 3   1.4023  1.2815     1    1     2    0.3429 1.2779 -0.0089  0.0000  1.2913  1.4462  1.5443



RUN NUMBER  99999              D E C I S I O N   M O D E L   R E S U L T S              SIMULATION TIME 4.00

INVESTMENT BATCH SIZE       50000.
ITEM SCALING CONSTANT          2.00
UNIT SCALING CONSTANT          0.80
UTIL. MEAS. UPPER LIMIT        0.00

UNIT     ITEM     ITEM      AUTH.   ON    DUE   DUE    UNIT    UTIL.    UTILITY    CUMUL.
 NO.      NO.     NAME       BAL.  HAND    IN   OUT    COST    INDEX    MEASURE     COST

  1    134.00  P  513J0        2     0     1     8     240.  -8.1322    5.1442      240.
  2    176.00  P  617A8        6     0     5    14    2000.  -4.8617    5.0097     2240.
  3     83.02  M  30134       14     2    11    58    9000.  -2.1199    4.9163    11240.
  4    134.00  P  513J0        3     1     1     8     240.  -8.1322    4.8877    11480.
  5     91.00  M  32251       22     5    16    33    9000.  -1.0513    3.2125    20480.
  6    314.00  A  13100        4     0     3     9   12000.  - .9177    3.1176    32480.
  7     84.01  M  30154       18     4    13     6    8000.  -2.0303    3.0019    40480.
  8    176.00  P  617A8        7     1     5    14    2000.  -4.8617    2.8671    42480.
  9    217.00  P  213E0        8     0     7    11    7000.  -1.1787    2.8650    49480.
 10    134.00  P  513J0        4     2     1     8     240.  -8.1322    2.2500    49720.
```

Fig. 24 -- Example outputs of the forecasting procedure and decision model

VI. POSTPROCESSOR

During simulation it is possible to generate a large amount of detailed data representing simulation results. The data may then be combined, summarized, and consolidated into output reports of various kinds. In many large simulation models, data generated during the simulation are processed afterward by a separate program called a postprocessor to obtain the summary reports.

In the Logistics Composite Model, the simulation program itself produces the main summary results as it operates. However, there is also a separate postprocessor that produces certain ancillary products. This program is further described here.

PURPOSE

Summary results produced during the simulation reflect the behavior and status of the simulated environment during discrete time intervals or at specific points in time in accordance with the reporting interval. To obtain results over the entire simulation, a number of output reports must be inspected. This is the function of the postprocessor in the L-COM model--to develop, in a single product, selected summary statistics over the entire simulation. In effect, it consolidates the periodic reports produced during the simulation. It also produces aircraft status information as a function of simulated time rather than at specific points in time as output during the simulation.

More specifically, the postprocessor produces two kinds of output products showing simulation results as functions of simulation time. These are the summary statistics and the aircraft displays, both in graphical form, which are described below.

SUMMARY STATISTICS

Preprocessor output consists of selected summary statistics that the computer plots in graphical form as functions of simulation time. Statistics produced in this manner are selected from among those contained in the performance summary report produced during the simulation. Thus, such summary statistics may be displayed as a percentage of sorties accomplished, average aircraft turnaround time, personnel utilization percentage, supply fill rate, equipment utilization percentage, and others.

Each statistic selected for display in this form is plotted on a separate page, as illustrated in Fig. 25. Although the graph for each statistic has simulation time as the abscissa, the ordinate may be scaled according to the particular statistic involved.

These graphs enable a user to discern how the simulated environment changes over time. They also help detect functional interactions such as the way in which aircraft NORS rates change as supply fill rates decrease or increase.

AIRCRAFT DISPLAYS

Another type of display the postprocessor generates is a plot, for selected aircraft, of the various tasks incurred during the simulation. This display shows not only tasks involving the airplane, but also shop repair tasks for components removed from the airplane. An example of this type of display is shown in Fig. 26. Each task is represented in a separate line, arranged in chronological sequence according to start time in decimal days. The duration of the task is represented by a contiguous sequence of alphabetic letters which also identify the kind of task. The meanings of these symbols are as follows:

Symbol	Definition
V	Service task
F	Actual flight
S	Aircraft scheduled maintenance
U	Aircraft unscheduled maintenance
B	Off-equipment actions
D	Depot actions
L	Imposed delays

These displays are useful in verifying that airplanes and components in the simulation do undergo processes that approximate real-world ones in terms of kinds and sequence of tasks. They also show visually how some logical functions of the model operate. For example,

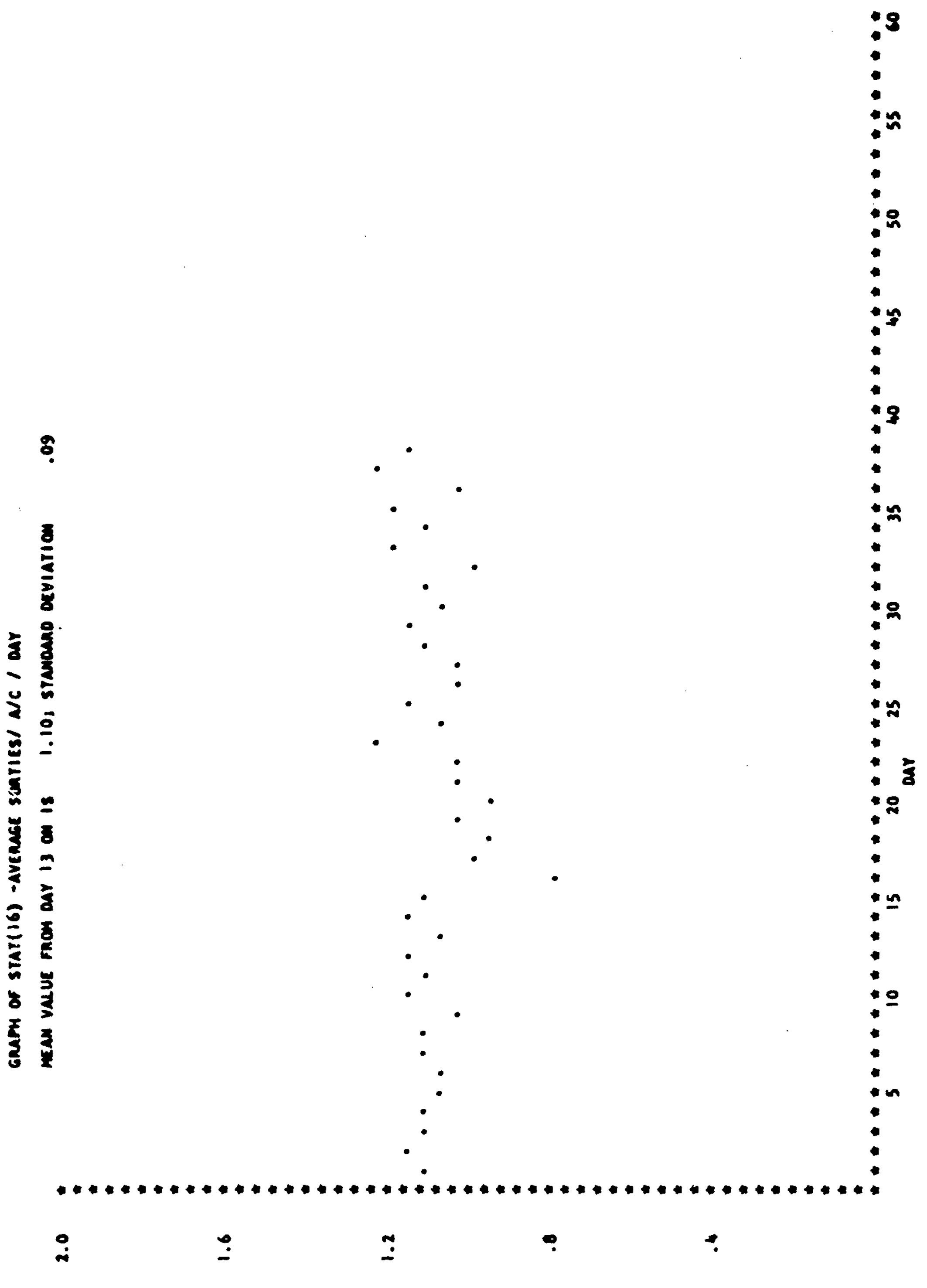


Fig. 25 -- Example summary statistic graph

```
                         A/C NO. 50887 DISPLAY ON DAY 10

.         .         .         .         .         .         .
3         4         5         6         7         8         9                  TASK     STOP    NO.
0123456789012345678901234567890123456789012345678901234567890123456789       NAME     TIME    PREM
   VVVVV                                                                     A01372   10.37   0
         FFFFFFF                                                             SORTIE   10.45   0
               VVVV                                                          A01310   10.48   0
                  SSSSS                                                      A03200   10.52   0
                   UUUUUU                                                    W75130   10.54   0
                        U                                                    Q75130   10.54   0
                        BB                                                   Y75130   10.55   0
                         BBBB       BB                                       F75130   10.68   1
                                       LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL       RCTIM    11.94   0
                    UUUU                                                     W13250   10.53   0
                       U                                                     Q1325A   10.53   0
                       BB                                                    Y1325A   10.54   0
                        DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD       N1325A   16.74   0
                     UUUUUU                                                  W14250   10.56   0
```

Fig. 26 -- Example aircraft display

BEST AVAILABLE COPY

if a task is preempted and finished at a later time, it will show up in the display as an interrupted sequence of symbols. The frequency and duration of such interruptions provide, at a glance, a general idea of resource scarcities at this point in simulation time. In general, however, these displays have limited value in deriving overall conclusions from the simulation, since they are produced only for a sample of airplanes, each being subject to random processes insofar as information in the display is concerned.

VII. OPERATING CHARACTERISTICS

To a large extent, the practical value of a simulation model is determined by how easy it is to use, the size of problem it can accept, the amount of computing time it requires, and, in some cases, how readily it can be transferred from one type of computer to another. In this section, the degree to which the Logistics Composite Model satisfies these practical considerations is discussed.

MODES OF OPERATION

In using L-COM, the first and most difficult step is to develop the inputs. Assuming data have not previously been established for this purpose, the data development program begins by constructing task networks that represent the processes of interest. The largest problem here is assessing the level of detail needed to obtain valid results. If too much detail is included, requirements for data collection, computer memory capacity, and computer running times can become excessive. If insufficient detail is included, valid results for the study objectives may not be obtained. The only general guidance we can offer is that the task networks should include as few tasks as possible--those which account for most of the resource utilization.

Once the task networks are established and appropriately labeled, further data for identified tasks can be obtained. As part of this data, resource requirements for each task must be determined. Having identified required resources, the remainder of the resource data may be collected. Next, data pertaining to the failure mechanism may be obtained according to relationships between tasks causing failures and affected resources. Following this, data for generating mission requirements may be established and initial entry points into the task networks can be identified. Finally, shift policies and associated data can be established, as well as decision model parameters, report specifications, and other miscellaneous data elements the model needs.

Having established an initial data base using the prescribed forms, the user may transcribe the data to punched cards and input it to the preprocessor. According to error messages and other preprocessor outputs, portions of the data may require correction and several preprocessor runs may be necessary before a valid set of data is obtained for use by the simulation program.

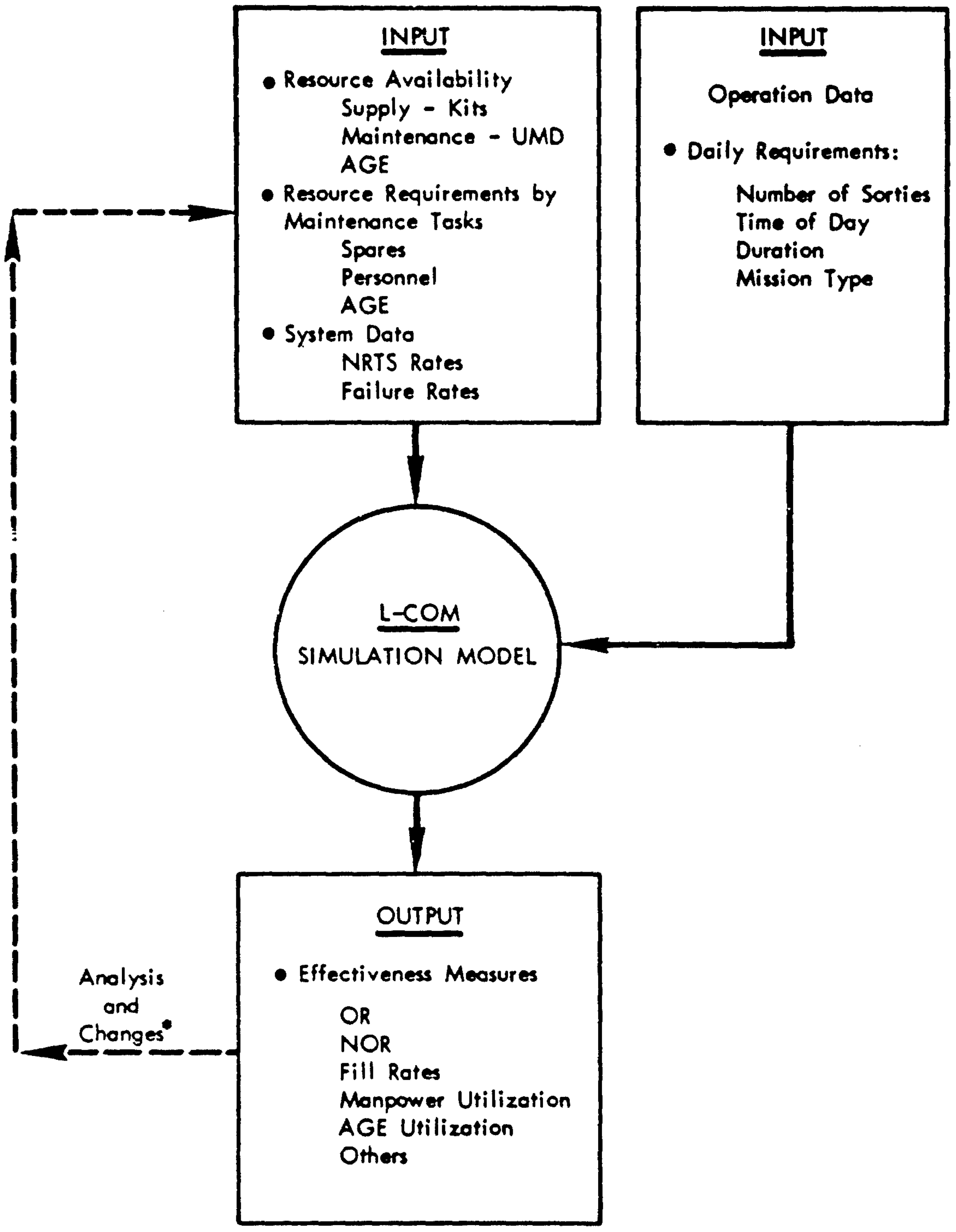
With the data base now established, the main simulation may be run a number of times, with portions of the data such as stock levels, task selection factors, or other policy parameters being changed between runs according to study objectives. This procedure is further illustrated in Fig. 27. For each run, options may be specified with regard to output products, the use of the forecasting procedure and decision model, and the use of the postprocessor functions.

According to overall study objectives, this procedure may be repeated several times for different data bases. Since only certain parts of the input data can be changed for different runs of the simulation model, changes in remaining portions will require a rerun of the preprocessor. Although the preprocessor does not require excessive computer time, the whole effort of making changes and running the preprocessor until a consistent and valid set of data is obtained can become considerable and should be avoided as much as possible.

The procedure outlined above represents the model's normal operating mode. For certain purposes, other modes are possible; for example, the preprocessor can be bypassed altogether by preparing input data directly in the required formats. Such operations, however, require complete familiarity with the programming aspects of the model.

COMPUTER REQUIREMENTS

The main computer requirements for the practical use of the L-COM model are that a SIMSCRIPT compiler must be available for the computer type being considered, and that it have available internal storage of at least the equivalent of 65,000 words, 36 bits in length. In addition to the UNIVAC 1107, upon which the model is currently operating, a variety of other computers may be used. As discussed below, the problems involved in changing from one computer type to another are relatively small, as long as the above criteria are satisfied.



*Both manual and automatic options available.

Fig. 27 -- Normal mode of operation for L-COM

MODEL CAPACITY

The largest problem that can be run on a given computer depends upon the capacity of its internal (core) memory, since the model is designed to operate totally in internal memory. Of the three parts of the model, the main simulation program has the most stringent memory requirements; it, therefore, controls the size of problem that can be run.

Using the UNIVAC 1107 for illustrative purposes, the allocation of memory for the simulation program is as follows:

Executive	8,000
SIMSCRIPT subroutines	7,500
Model program	
Definitions	2,200
Logical subroutines	12,500
Reports	5,800
Available for data	29,000
	65,000

The 29,000 words of storage available for data may be further divided into two parts, one for initialization data and the other for dynamic and working data. The amount of storage required for initialization data depends largely upon the following four factors:

T	Number of tasks in task networks
A	Average number of resources per task
R	Number of different resources
C	Number of columns on Performance Summary Report

Based upon these factors, the following formula may be used to estimate memory requirements for initialization data:

$$\text{Initial Data} \simeq (5.55 + A)\ T + 6.75\ R + 71\ C + 1000.$$

The use of this formula may be illustrated by an example. Suppose that there are 1000 tasks, each requiring three types of resources on an average. Suppose further that there are 500 different kinds of resources altogether and that there are 10 columns specified for the Performance Summary Report. Inserting these values in the formula, the following results are obtained:

$$\begin{aligned}\text{Initial Data} &\simeq (5.55 + 3)(1000) + (6.75)(500) + (71)(10) + 1000\\ &\simeq 8{,}550 + 3{,}375 + 710 + 1000\\ &\simeq 13{,}635\end{aligned}$$

Using this result, about 15,400 words of memory would be left for dynamic storage requirements and working space, plus 2500 words that become available after the initial data are established. This would probably be sufficient, but whether or not it is enough cannot be determined in advance. In general, this kind of memory requirement depends upon frequency and patterns of mission requirements, resource levels, policy parameters, and other factors, all of which interrelate in an exceedingly complex fashion in determining how much storage is needed.

From this analysis, it can be seen that the maximum size problem that can be run on a given computer cannot be precisely established in advance. As indicated by the formula for initialization data memory requirements, there are tradeoffs among the four main factors involved. However, the formula and judgments concerning dynamic storage requirements may be used to estimate whether or not a given problem will fit on the computer. The formula, however, is only valid for the UNIVAC 1107 due to the way data are packed; it must be appropriately altered for any other computer type.

RUNNING TIMES

The computer time required to run the L-COM model depends, of course, upon the size of problem and type of computer used. For the UNIVAC 1107, we have gained sufficient experience to estimate running times for various size problems. The most useful factor for estimating simulation running times is the number of tasks that are simulated per minute of computer time. This factor generally lies between 800 and 1000 tasks per minute, depending upon how much back-ordering occurs due to resource scarcities. With this factor, estimates may be made of running times for various flying programs.

As an example, assume that a four-squadron base is simulated, each squadron having 18 airplanes that average one sortie per airplane per day. Assume further that task networks are defined such that 20 different tasks are averaged per sortie. Thus, about 1540 tasks per day at the base would be simulated; using the above planning factor, a day's worth of operations would therefore take 1-1/2 to 2 minutes of computer time. For this example, we might generalize that to simulate operations of a full base with a reasonable level of detail and with a standard flying program would require somewhere between 45 and 60 minutes of computer time for the UNIVAC 1107 for thirty days of simulation.

Extrapolating this to other examples, however, is quite hazardous. But if an estimate can somehow be made about the average number of daily tasks at base, the planning factor of 800 to 1000 tasks per minute of simulation may be used to estimate total running time for a UNIVAC 1107 or other machine of equivalent speed.

In addition to running times for the main simulation program, the preprocessor and postprocessor also require time to run. Although these running times depend upon how large the data base is and how much processing is done, estimates for maximum running times can be made from previous experience. For a data base large enough to approach the capacity limit of the computer, the preprocessor requires about 20 minutes; a typical mix of postprocessor outputs requires about 10 minutes. Again, these figures pertain to the UNIVAC 1107 running times and must be correspondingly adjusted if another type of computer is considered.

TRANSFER TO OTHER COMPUTERS

As previously mentioned, the Logistics Composite Model is now operating on the UNIVAC 1107 at Wright-Patterson Air Force Base, Ohio. If another type of computer were to be used, a SIMSCRIPT compiler would have to be available for that type of computer. Fortunately, such compilers are available (or about to become available) for practically all computers large enough to handle the model described here.

At present, there are two versions of SIMSCRIPT: I and I.5. If the model were transferred to a computer that has a SIMSCRIPT I.5 compiler, little difficulty would be encountered in the conversion.

There are two short machine language routines in the preprocessor which allow for the variable format of entries on the input forms. Essentially, a programmer would need a few days to rewrite these programs, change control cards, and make other minor adjustments necessary for operation on the new type of computer. If only a SIMSCRIPT I compiler were available, the conversion time would be somewhat greater, since a number of program changes would be needed, particularly in the preprocessor and postprocessor. Altogether, several weeks of programming effort would be needed.

VIII. MODEL VALIDATION

A valid simulation model requires a structure that adequately represents the environment so that results will be consistent with those of the real world. This section discusses verification and validation procedures for the Logistics Composite model.

VALIDATION REQUIREMENTS

Since it is impossible to capture all real-world operations in a simulation model, various abstractions, simplifications, and omissions must be made. These compromises are reflected in both the input data and the structure of the model itself. In using the model to investigate system behavior, the user must be sure that the compromises he makes can be explained and do not significantly affect the results. Without making such allowances in interpreting results, any conclusions derived would be suspect. A variety of techniques have been developed for accomplishing the model verification and validation functions.*

With respect to verification requirements, the Logistics Composite Model is somewhat unique in that most of the structure of the environment is represented in input data the user provides. In constructing task networks, the user has direct control over what portions of the base environment are included in the simulation and the level of detail in which they are to be represented. To this extent, the verification requirement becomes the user's responsibility. He must satisfy himself that he has captured as much of the environment, in terms of scope and detail, as is consistent with his study objectives. This may be accomplished, in general, by modifying and adjusting task network and other data until results are obtained that are acceptable for his objectives.

*A discussion of such techniques is contained in G. S. Fishman and P. J. Kiviat, *Digital Computer Simulation: Statistical Considerations,* The RAND Corporation, RM-5387-PR, November 1967.

In addition, there are significant features of real-world operations that the current version of the model does not allow for; some of these features are identified in Sec. IX for inclusion in later versions. The omission of these features can significantly affect results and must be considered or compensated for in interpreting the results. Because of these omissions, model results must be appropriately tempered and qualified.

Insofar as validation of the model is concerned, the main criterion is being able to reproduce corresponding measures obtained from real-world experience and/or to explain satisfactorily any differences that might exist. The largest problem in this procedure is to determine the proper level of detail necessary to satisfy the validation criterion. In constructing task networks with associated resource definitions, it is desirable to minimize the number of different kinds of tasks in order to conserve computacion time and memory requirements; however, the validation criterion cannot be satisfied if task representations are too highly abstracted. There is no easy solution to this problem, and recourse must normally be taken to trial and error procedures tempered by judgment and experience.

USE OF PROJECT PACER SORT DATA

To validate the Logistics Composite Model, we plan to use experience and data obtained from the field test portion of the PACER SORT project. A comprehensive data collection effort included in the field test facilitates its use for this purpose.

For the validation, the actual flying program experience of the field test will be input to the model. Task networks for the aircraft and major components included in the test have been constructed. Resource levels will be input according to those established in the test. In general, as much of the field test environment as is practical will be represented in the inputs. With this data, the main validation objective will be to determine how well the model results compare with those experienced in the field test.

For use in the model, task networks for the F-4C aircraft and major components have been developed. Approximately 80 subnetworks like that in Fig. 28 have been established, containing 1700 different tasks. Associated with these are 503 different kinds of resources, broken down as follows:

Aircraft types	4
Personnel types	21
AGE items	128
Reparable components	350

The four aircraft types are used to represent the four squadrons of F-4C aircraft included in the test. The 350 reparable components and associated personnel and AGE items were selected because they account for most of the maintenance accomplished during the test; these items either have relatively high failure rates or require extensive repair when they fail. With these data, a number of model runs will be made for validation purposes.

In general, validation requirements will be considered largely met if the model produces results matching those experienced in the field test for the following critical performance measures:

Percentage sorties accomplished
Average aircraft turnaround time
Average maintenance manhours per flying hour
Personnel utilization percentage
AGE utilization percentage
Not-operationally-ready supply (NORS) rate.

If the model produces comparable results for all these measures simultaneously, there will be considerable assurance that functional interactions in the simulation resemble those of the real world. If there are significant differences in one or none of these measures, however, explanations must be sought, either in the content of the input data or in the structure of the model, particularly insofar as omitted features might be involved.

As part of validation, the task network descriptions are subject to considerable change because of intermediate model results. In general, these changes involve variations in task arrangements and the addition of certain tasks to more fully account for utilization of available resources. Also, factors such as task durations, failure data, and stock levels are being refined during the validation process.

BEST AVAILABLE COPY

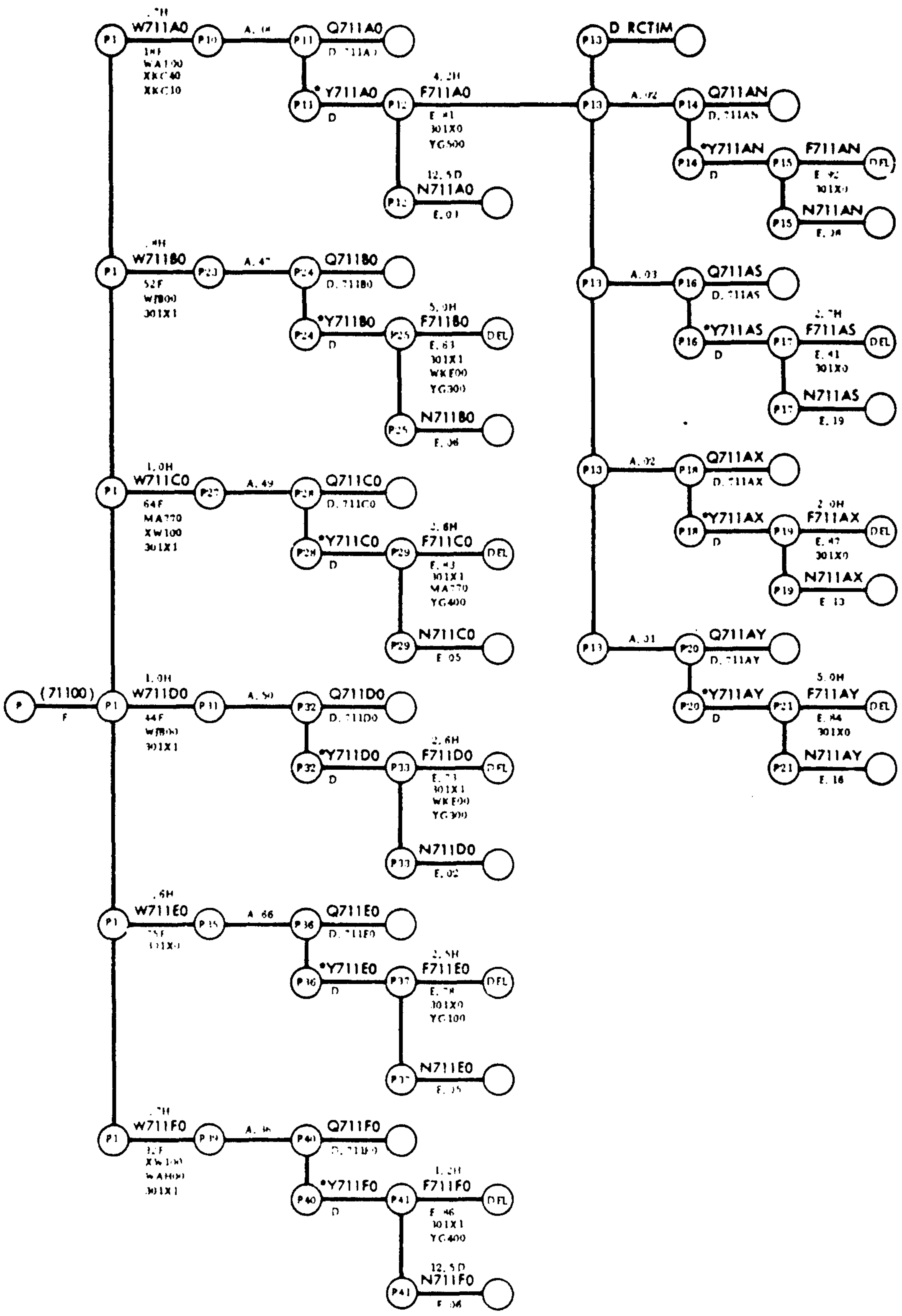


Fig. 28 -- Example network data for F-4C radio navigation system

In most cases, these changes, extensions, and refinements are made to correct original deficiencies in the data base as found during the validation. Since the structure of the simulated environment is largely specified by the input data, the iterative procedure of altering the structure to improve intermediate results of the validation criteria represents a somewhat unique approach to validation. Despite this kind of flexibility, however, it is expected that results based upon PACER SORT experience will not fully satisfy validation requirements due to omissions of certain real-world features in the model's structure.

Another limitation of the validation is the extent to which data collected during the field test adequately represent the actual environment. For example, if the data misrepresent the supply availability that actually existed, then validation discrepancies could occur for supply performance measures, even though the model correctly replicates the true performance. In validating the model, therefore, judgments must be made concerning the overall adequacy and accuracy of the reported field test results which the model results are compared against.

In view of these limitations and others, only broad validation objectives can be achieved from Project PACER SORT experience. At the very least, however, these objectives include indications that model results are not nonsensical, that they are internally consistent among the logistics functions, and that they are reasonable approximations of the experienced activity. At later times, the model may be further validated with other sets of data.

VALIDATION RESULTS

Preliminary validation results based upon Project PACER SORT experience indicate that broad validation objectives for the L-COM model can be satisfied. The extent to which they are satisfied, however, will depend upon final results. When the validation program is completed, a separate report is planned describing the program and results in further detail.

IX. FUTURE EXTENSIONS AND APPLICATIONS

The Logistics Composite Model described here is the initial version (MOD 1) that is currently programmed and operating. It has several limitations that either constrain the scope of its application or fail to capture and represent features of the simulated environment. Several extensions and refinements are planned to alleviate these limitations. They are discussed in this section.

STRUCTURE OF FUTURE VERSIONS

In the future, we plan to expand the overall structure of the model to include two new major features--a repair level decision model and a data bank. Figure 29 illustrates how these components will relate to the current version of the model.

The main function of the repair level decision model is to select, from the many thousands of parts and other resources involved in the support of a weapon system, those to include in the simulation process. Resources having the most impact upon weapon system performance will be selected for detailed representation in the simulation; remaining resources will be aggregated and abstracted.

The data bank will accrue a reservoir of input data such as information on a number of weapon systems, operating environments, and other factors. This will be stored in a large volume memory device, such as a disk unit or a drum, so that data needed for a particular application of the model can be readily extracted.

In addition to these new elements, extensions and refinements in the main simulation programs and embedded decision model are planned in order to increase the validity of the results by more adequately representing the environment and decision processes included in the simulation.

Some of the new features of the model may be inserted in the version now operating. Others, however, must await the availability of a computer with a larger capacity and computing power than the one currently being used.

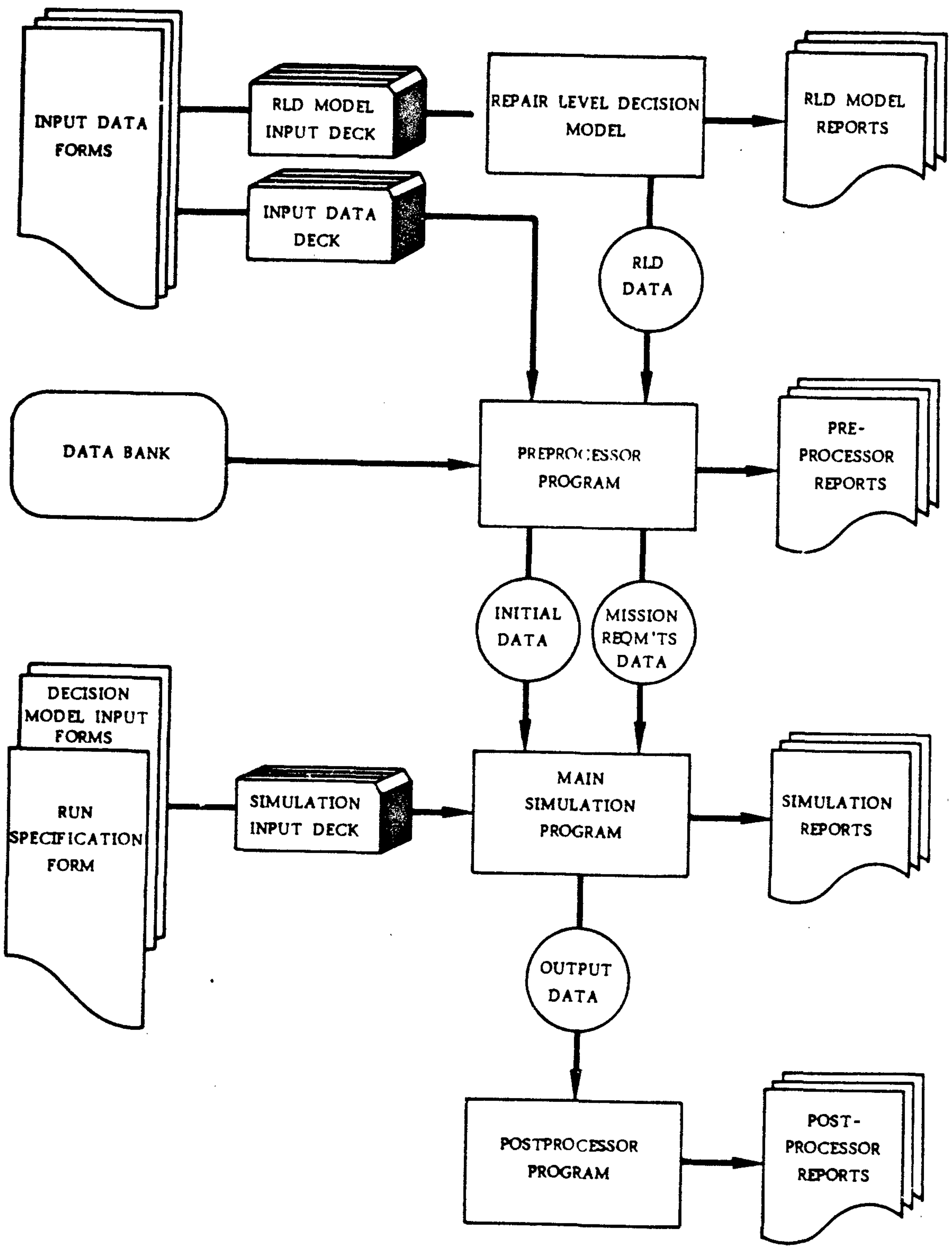


Fig. 29 -- Overall structure of Logistics Composite Model (MOD II)

REPAIR LEVEL DECISION MODEL

A weapon system consists of thousands of parts and is supported by hundreds of different kinds of resources such as maintenance personnel of various skills, ground support equipment, facilities, and so on. Because of computer limitations, it is impossible to represent all of these resources and their interactions in the simulation of operations, maintenance and supply functions. Actually, it is seldom necessary to include all of this information since some of it does not significantly contribute to the accuracy of the results. The problem is choosing which portions of the weapon system and supporting environment to represent in some detail and which to abstract or omit altogether.

Another problem concerns which repair policy to assume. A particular repair policy must be prescribed to operate the model. The policy affects the factors assigned for task selection and the construction of the task network. For an existing weapon system, a repair policy has already been established and task networks can be constructed to reflect this policy. If it is desired to investigate postulated alternatives to the established repair policy, the network can be correspondingly changed and the simulation model can be used to evaluate the consequences. For a new weapon system, however, determining an appropriate repair policy might be a major study objective. For both existing and new weapon systems, it might be desired to find a preferred or best repair policy under a number of constraints and to then evaluate this policy by simulation. Determining this policy and the corresponding task network needed for input to the simulation represents a significant problem.

The repair level decision model represents an analytic approach designed to help solve these problems. As input data, the hierarchical structure of the weapon system is described, both in terms of the physical breakdown of included parts and 'n terms of steps taken to isolate faults that might occur. Also input are reliability factors for included parts and various cost factors relating to repair and supply functions involved in weapon system support. Based upon these data, the model evaluates possible combinations of repair actions that might be taken when a particular item fails, and determines the combination which will minimize expected overall support costs. As outputs, the model specifies whether or not an item should be repaired when it fails and, if repaired, whether it should be done by organizational (flight-line) maintenance, base repair shops, or depot repair facilities. As a by-product of this decision process, the model also determines and generates the requirements for maintenance personnel, supply levels and AGE items that correspond to the least-cost repair level decisions.

This model identifies the items contributing most to the total expected support cost. For a typical weapon system, a few hundred items account for almost all of the repair and supply support costs that are incurred. Once the initial range model identifies the troublesome parts, they can be explicitly represented in the simulation, with the remaining items being represented in aggregate form. Also, the repair level decisions made by the initial range model enable corresponding task networks to be developed for use in the simulation.

The repair level decision model may be used independently to analyze and determine preferred repair policies for weapon systems. However, results are based upon minimizing expected values of future costs a in obtaining these results, a variety of operational factors are n considered. For example, the effects of dynamic flying programs, the use of substitute items, cannibalization practices, task preemption, and other aspects of the operational environment are not represented in the model logic. For this reason, the simulation model which recognizes these factors is used to evaluate and further refine decisions produced by the initial range model. In general, the initial range and simulation models are designed to complement and reinforce each other in determining preferred support systems.

Experimental versions of the repair level decision model have been constructed and are being tested. At an appropriate point in its development, the program will be incorporated as an integral part of the overall L-COM model as suggested above. A separate report describing the repair level decision model and its use is planned.

ADDITIONAL SIMULATION FEATURES

Several additions to the simulation portions of the L-COM model are possible to represent certain features of the operational environment. Some of these are as follows:

1. *Conflicting Maintenance*. Although the task network can be used to identify aircraft maintenance processes that can be accomplished simultaneously, conflicting maintenance constraints upon parallel tasks exist which depend upon the particular tasks selected. Since tasks are selected randomly, the constraints cannot be reflected in the task network as currently defined. The constraints are of two main types.

One type is a physical limit on the number or type of resources used by several tasks occurring simultaneously. An example might be several tasks that involve work in the airplane's cockpit. Since the working space is extremely limited, only one or two technicians can be employed, and the required tasks must be accomplished sequentially rather than in parallel.

Another type of conflicting maintenance is a functional one where two tasks performed simultaneously might create a hazardous situation. A typical example is where work must be done on the fuel tanks and the electrical power system of an airplane. Both tasks cannot be done at the same time because an electrical spark might touch off a fuel explosion. These tasks must therefore be done in sequence, although which is done first is immaterial.

Including these constraints in the simulation will more realistically represent actual maintenance practices, as well as improve accuracy of results such as aircraft turnaround times and personnel utilization rates.

2. *Shop Cannibalization*. In the current version of the simulation model, cannibalization is represented only to the extent that needed parts are borrowed from other airplanes that are in maintenance.

Another level of cannibalization consists of borrowing needed parts from other components or assemblies awaiting repair in the base repair shops. Such cannibalization may even extend to the borrowing of parts from test equipment or from serviceable higher assemblies in base supply. In some cases, this type of cannibalization can significantly affect system performance. When this feature is included in the simulation, the effects and consequences can be measured.

3. *Item Criticality*. When certain items fail and cannot be replaced due to lack of spares, sometimes the airplane can still be used for some types of missions. For example, a malfunction of the fire control system might not prevent using an airplane for a cross-country training flight. Other item failures, such as in the communications system, may ground the airplane until the malfunction is corrected.

In the current model, all items are considered critical in the sense that their failure will prevent further use of the airplane until fixed. Adding item criticality considerations to the model will represent deferred maintenance and will produce more accurate results with respect to mission performance.

4. *Weather Effects*. Weather very often has an important effect upon base operations. Airplanes may be prepared for a mission and then be unable to take off because of adverse weather at the base or in the target area. In the current model, weather effects can only be implicitly considered by the way in which mission requirements are generated. A more explicit representation will enable the impact of weather factors upon overall base operations to be measured and analyzed.

5. *Multibase*. The current L-COM model only considers aircraft operations at one base. Actually, the definition of a base in this context is somewhat ambiguous. Insofar as the operation of the model is concerned, a base consists of any common pool of resources supporting tasks, defined by a task network, generated in accordance with mission requirements. Although this description fits that of an ordinary base, it may also pertain to more unusual situations such as certain types of dispersed operations.

Later versions of the model may include explicit recognition of

BEST AVAILABLE COPY

aircraft operations at a number of bases simultaneously. When this feature is added, certain interactions can be represented. For example. several kinds of lateral support can be simulated, where one base provides needed resources to another. As another example, procedures for allocating mission requirements among bases according to their aircraft availability may be inserted in the simulation. Transportation functions and base-depot interactions can be better represented in a multi-base simulator.

Although there appear to be no difficult conceptual problems involved in extending the model to the multibase situation, significant computation and machine capacity problems may occur. Unless the number of resources and the size of the task network are severely curtailed, a multibase simulation will require a considerably larger and faster computer than the one currently being used.

DECISION MODEL REFINEMENTS

The forecasting procedure and decision model currently included as subroutines in the L-COM model are relatively crude and subject to refinement in later versions. Since the forecasting procedure is based upon an exponential smoothing technique, it can react only to past data observed in the simulation. For sudden changes in the flying program, it often reacts too late If the smoothing constant were raised to increase the response time, the procedure would overreact to random fluctuations in the performance measure. For these reasons and others, a better forecasting procedure, perhaps one using some kind of regression technique based upon future flying programs, may be developed and applied in place of the present procedure.

The decision model is also subject to considerable improvement. Preliminary experience shows that the way in which the utilization index is defined is not completely satisfactory, and some better measure for resource posture relative to requirement may be developed. Also, in considering different times for augmentation, the model assumes that they all have equal impact upon the performance measure being used. Since this is not usually true, some way to reflect the difference might be developed, perhaps based upon the remoteness of the resource from the airplane in its use.

Another major deficiency is that the decision model only increases resource levels and does not decrease them. Thus, it does not recognize and appropriately reduce surpluses that might occur. If this deficiency is corrected and the simulation extended to a multibase environment, the decision model may then be used to allocate resources among the various bases effectively.

DATA BANK

The development of data required for input to the Logistics Composite Model, particularly that pertaining to the task networks, is not a very easy job. Once developed for a particular weapon system or aircraft type, however, the data may be used for a variety of studies. As the model is applied to different weapon systems and operating environments, it is envisioned that the corresponding data might be accumulated in a systematic fashion into a data bank from which information needed for a particular study could be readily extracted.

Such a data bank might include the following classes of information:

1. Flying programs for possible contingency deployments
2. Task networks and associated data, in several levels of detail, for various types of weapon systems and repair policies
3. Factors for alternative resource leveling procedures and shift policies

If a data bank containing this type of information in machine-accessible form is established, it might be possible to provide a user with a simple check list or questionnaire-type form for specifying a particular simulation.* Thus, he might identify a particular

*A similar technique is used in A. S. Ginsberg, H. M. Markowitz, and P. M. Oldfather, *Programming by Questionnaire*, The RAND Corporation, RM-4460-PR, April 1965.

type of flying program, a generic description of a repair policy and task network for a particular aircraft type, and a particular procedure for computing resource levels. According to these specifications, the computer could then extract the necessary detailed data from the data bank and use it for the simulation.

From a programming point of view, the data bank concept would have its main impact in the preprocessor functions. In general, the preprocessor as currently developed would be replaced by an information retrieval program designed to extract needed data from the data bank according to user-provided specifications.

ON-LINE SIMULATION

In the present version of the model, a user must wait until the entire simulation is finished before he can view the results. In a later development, a concept of on-line simulation may be applied wherein a user may see results as they generate during the simulation. Furthermore, methods can be provided for enabling a user to interact with the simulation by altering certain factors according to intermediate results, thereby changing the direction of the subsequent simulation. For example, intermediate results may indicate a need for altering the overtime policy; by changing the relevant parameters, the user can alter the overtime policy in subsequent operation of the simulation. Similarly, this feature enables a user to experiment with various policy parameters wherein impacts can be immediately discovered as the parameters are changed.

For the on-line mode of simulation, visual display devices such as a CRT (cathode ray tube) device would be used. Many results would be presented in graphical form similar to those the postprocessor generates. As the simulation proceeds in the computer, the graphs would simultaneously be generated and displayed. Procedures would be provided for rapidly selecting desired displays and for altering values of policy parameters or other portions of the data base. When results are obtained that need to be preserved, copies of the displays in more permanent form can be made.

This concept, when applied to the Logistics Composite Model, will enable a user to view the overall operation of a base in accelerated time. Within a few minutes, several days' operations can be witnessed. Functional interactions, resource utilizations, and other aspects of the operating environment can be more easily identified. In general, on-line simulation can constitute a valuable technique for use in a variety of planning studies.

MAJOR APPLICATIONS

The preceding discussion has suggested a number of uses and application areas for the Logistic Composite Model. There are still two main applications of special importance, particularly with respect to the use of advanced versions. One is the determination of logistics requirements in support of contingency operations; the other is the determination of preferred repair policies and associated resource requirements for new weapon systems.

A contingency deployment is characterized by a dynamic flying program which heavily stresses the supportive logistics system. In many cases, as many sorties as possible must be flown, the number being limited solely by the ability of the support system to turn around airplanes after each sortie. Determining this maximum flying capability and the associated requirements for support resources constitute an important but difficult problem. The problem is compounded by the number of different kinds of contingency deployments that might be considered in terms of operating environment, dynamic factors involved, and weapon systems used. The L-COM model is particularly useful in studying these deployment problems because it permits a full representation of the major facets of the overall support system involved and their interactions in meeting the mission requirements. Also, the decision aspects of the model help determine the resource posture needed to best support the deployment. In many respects, the use of the model provides a viable alternative to the field tests or other, less accurate analysis techniques that might otherwise be required for such purposes.

In modern weapon system design philosophy, logistics support is viewed as a part of the overall weapon system, materially affecting the design of the weapon itself. Lifetime support costs become an important ingredient in the design criteria, where many components may be designed that have higher initial costs but will have significantly lower subsequent repair and supply costs. The determination of break-even points in this regard is a difficult problem because of the functional interactions involved. Repair policies, supply and maintenance personnel requirements, and design characteristics of ground support equipment are all interrelated in this problem. Advanced versions of the Logistics Composite Model can become a valuable analysis technique for these kinds of problems.

The repair level decision model, as previously described, can be used to make preliminary determinations of a preferred support system in terms of repair policies and associated resource requirements. These initial determinations may then be refined by using the simulation and embedded decision models. In this regard, the task network concept of the simulation program is particularly useful, since it enables components subject to design review to be represented in necessary levels of detail. By using these features of the L-COM model interactively in the design of a weapon system, tradeoffs can be analyzed, leading to an optimization of the overall weapon system including logistics support.

Although these two application areas are identified as especially significant insofar as future use of the L-COM model is concerned, there are many other study areas and problems in which the model can serve a role. In general, any problem involving significant interactions among the many functions accomplished at an Air Force base can be analyzed by this technique.

Appendix

GLOSSARY OF DATA ELEMENTS

This Appendix lists and defines the data elements included in user-provided input forms for the Logistics Composite Model. The data elements are grouped according to the following forms:

Form Number	Title
1	Run Specifications
10	Performance Summary Report Specifications
11	Task Network
12	Task Definitions
13	Resource Definitions
14	Failure Clock Decrements
15	Distributions
16	Shift Change Policy
17	Mission Entry Points
20	Sortie Generation Data

In addition to the listed data elements, each line on the forms contains a preprinted card number that identifies the corresponding data when converted to punched card form.

The above list includes all input forms required by the model except for Form 18 (Priority Specifications), Form 30 (Forecast Parameters Specifications), and Form 40 (Decision Model Factors). Data elements on these two forms are excluded from this Appendix because they are defined on the forms themselves.

FORM 1--RUN SPECIFICATIONS

Model Selection	An "X" mark in the indicated space that causes the forecasting procedure and/or the decision model to operate.
Report Selection	Entering simulation times in the spaces provided causes production of the associated special reports at certain times.

FORM 10--PERFORMANCE SUMMARY REPORT SPECIFICATIONS

Reporting Cycle	The time between successive productions of the Performance Summary Report, Level I.

(Preceding Pg. Blank)

Number of Cycles	The number of Level I reporting cycles before Level II of the Performance Summary Report is produced.
Number of Groups (Categories)	The number of columns, for respective parts of the Performance Summary Report, in which summary statistics are to be displayed.
Column Headings	Labelings to appear over each column of the Performance Summary Report for respective parts of the Report.

FORM 11--TASK NETWORK

Prior Node	The label for the beginning node of a task in the task network.
Selection Mode	A code identifying and controlling whether or not the task will be incurred when encountered.
Task I.D.	A label identifying the task in the task network.
Next Node	The label for the ending node of a task in the task network.
Selection Parameter	A factor representing the probability of selecting the task when encountered.
Task Description	A verbal description of the task.

FORM 12--TASK DEFINITIONS

Task I.D.	The identification of the task as established on Form 11.
Priority	The priority code assigned to the task.
Task Duration Mean Variance Distribution Type	The length of time a task will last; selected as a random draw from a probability distribution of the specified type, with a mean and variance as indicated.
Associated Resource	An identification of the major resource with which the task is associated.
Resource Rqrs Resource Quantity	Identifications of the kinds and quantities of resources needed to accomplish the task.

FORM 13--RESOURCE DEFINITIONS

Resource I.D.	The identification of the resource as established on Form 13.

Resource Type	A code identifying the type of resource (aircraft, personnel, parts, AGE, facilities).
Report Column	The column in the Performance Summary Report into which statistics for the resource are to be accumulated.
Unit Cost	The per unit acquisition cost of the resource.
Authorized Quantity	The number of units of the resource authorized for stockage.
Substitute Resource	The identification of a substitute resource if any.

FORM 14--FAILURE CLOCK DECREMENTS

Task I.D.	The identification of the task with which resource failures are associated.
Mode	A code identifying the failure mode (flying hours, number of landings, etc.) for associated resources.
Resource I.D.	For each task causing failures, the identification of associated resources subject to failure.
Decrement	For each resource subject to failure, the amount by which its failure clock is reduced each time the associated task is incurred.

FORM 15--DISTRIBUTIONS

Distribution I.D.	The identification of a particular probability distribution.
Type	A code for type of distribution.
Format	A code for format of the distribution data.
Probability	The discrete or cumulative (depending on distribution type) probability of occurrence for the corresponding value.
Value	The value of a variable to which the corresponding probability applies.

FORM 16--SHIFT CHANGE POLICY

Code	A code identifying the type of data entered in the corresponding row.
Resource I.D.	The identification of resources subject to the shift policy.

Shift Durations/ Authorized Levels	The length of each shift (if code = *), the number of shift cycles (if code = R), or the authorized level (if code = blank).

FORM 17--MISSION ENTRY POINTS

Mission I.D.	The identification of the mission type.
Report Column	The column in the Performance Summary Report into which statistics for the mission type are to be accumulated.
Network Entry Point	The identification of the starting node in the network for aircraft assigned to the mission.

FORM 20--SORTIE GENERATION DATA

Day	The simulation day in which the mission is to occur.
Number of Missions	The number of missions of the same type to be accomplished, or the identification of a probability distribution from which the number is to be drawn.
Takeoff Time	The takeoff time for the mission, or the identification of a probability distribution from which the number is to be drawn.
Aircraft I.D.	The type of aircraft required for the mission.
Mission I.D.	The type of mission.
Mission Size Minimum Maximum Spare	The number of airplanes needed for the mission in terms of a minimum, maximum, and number of spares.
Mission Length Mean Variance Distribution Type	The total flight time for the mission as drawn from a distribution of the indicated type, mean, and variance.
Lead Time	The time prior to takeoff needed to prepare aircraft for the mission.
Cancel Time	The time after takeoff time at which a mission is canceled if required aircraft are unavailable.
Cycle Interval Stop	The time interval over which the indicated missions are to be repeated until the indicated stop time is reached.

BEST AVAILABLE COPY

DOCUMENT CONTROL DATA

1 ORIGINATING ACTIVITY THE RAND CORPORATION	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP
3. REPORT TITLE THE LOGISTICS COMPOSITE MODEL: AN OVERALL VIEW	
4. AUTHOR(S) (Last name, first name, initial) Fisher, Captain R. R., W. W. Drake, J. J. Delfausse, A. J. Clark and A. L. Buchanan	
5. REPORT DATE May 1968	6a. TOTAL No. OF PAGES 114 6b. No. OF REFS. ---
7 CONTRACT OR GRANT No. F44620-67-C-0045	8. ORIGINATOR'S REPORT No. RM-5544-PR
9a AVAILABILITY/LIMITATION NOTICES DDC-1	9b. SPONSORING AGENCY United States Air Force Project Rand

10. ABSTRACT

A description of the Logistics Composite Model (L-COM), a computer model developed to simulate the overall operations and support functions at an Air Force base. L-COM consists of three main programs: a preprocessor, a simulation program, and a postprocessor. The model replicates the flying of aircraft; accomplishment of servicing tasks; incurrence of malfunctions; flight-line maintenance; repair of components in base repair shops; the utilization and interaction of resources in the demand process; and the changes in resource availability according to shift policies. L-COM has two unique features: (1) a task network that describes base processes to be simulated by identifying particular tasks and the sequence for accomplishing them; and (2) embedded decision routines that help determine a best mix of resources to support a prescribed flying program. The model requires a computer with an internal memory of at least 65,000 words of 36-bit length or equivalent. A typical problem requires from 1 1/2 to 2 minutes of computer time to simulate a day's worth of base operations involving 1500 tasks. Since L-COM is written in SIMSCRIPT, almost any computer of sufficient size may be used.

11. KEY WORDS

Logistics
Models
Maintenance
Computer Simulation
Data Processing
Bases
PACER SORT (Project)